

Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/JP05/007540

International filing date: 20 April 2005 (20.04.2005)

Document type: Certified copy of priority document

Document details: Country/Office: JP
Number: 2004-133102
Filing date: 28 April 2004 (28.04.2004)

Date of receipt at the International Bureau: 24 June 2005 (24.06.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)



World Intellectual Property Organization (WIPO) - Geneva, Switzerland
Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application: 2 0 0 4 年 4 月 2 8 日

出 願 番 号
Application Number: 特 願 2 0 0 4 - 1 3 3 1 0 2

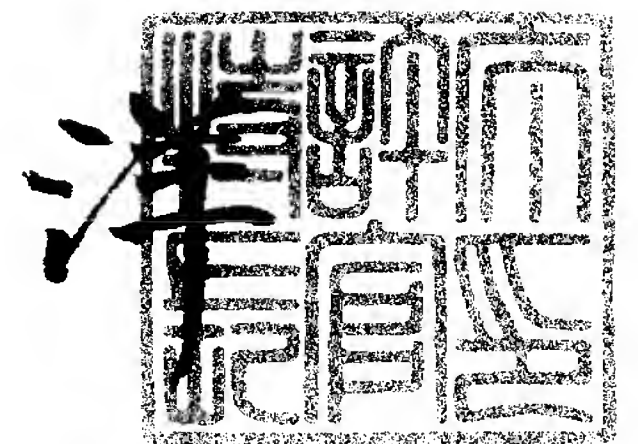
パリ条約による外国への出願
に用いる優先権の主張の基礎
となる出願の国コードと出願
番号
J P 2 0 0 4 - 1 3 3 1 0 2
The country code and number
of your priority application,
to be used for filing abroad
under the Paris Convention, is

出 願 人
Applicant(s): 松下電器産業株式会社

2 0 0 5 年 6 月 8 日

特許庁長官
Commissioner,
Japan Patent Office

小 川



【書類名】 特許願
【整理番号】 2048260080
【提出日】 平成16年 4月28日
【あて先】 特許庁長官殿
【国際特許分類】 G06F 1/04 301
【発明者】
 【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内
 【氏名】 天野 克重
【発明者】
 【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内
 【氏名】 水山 正重
【特許出願人】
 【識別番号】 000005821
 【氏名又は名称】 松下電器産業株式会社
【代理人】
 【識別番号】 100097445
 【弁理士】
 【氏名又は名称】 岩橋 文雄
【選任した代理人】
 【識別番号】 100103355
 【弁理士】
 【氏名又は名称】 坂口 智康
【選任した代理人】
 【識別番号】 100109667
 【弁理士】
 【氏名又は名称】 内藤 浩樹
【手数料の表示】
 【予納台帳番号】 011305
 【納付金額】 16,000円
【提出物件の目録】
 【物件名】 特許請求の範囲 1
 【物件名】 明細書 1
 【物件名】 図面 1
 【物件名】 要約書 1
 【包括委任状番号】 9809938

【書類名】 特許請求の範囲

【請求項 1】

動作周波数を変更できるCPUを備えた情報処理装置において、

実行するプログラムの実行時間に関する要求と、前記プログラムの実行時間に関する要求の許容範囲を登録することの出来るプログラム実行時間登録部と、

前記プログラムの実行に必要な処理量を検知するプログラム処理量検知部と、

前記プログラム実行時間登録部によって登録された実行時間に関する要求と許容範囲を満たす範囲で、前記プログラム処理量検知部が検知したプログラムの実行に必要な処理量の単位時間毎の合計が均等化するように、前記プログラムの実行時間を決定するプログラム実行時間決定部と、

前記プログラム実行時間決定部により均等化された単位時間毎の必要な処理量に基づき、前記CPUの動作周波数を決定するCPU動作周波数設定部と、

前記プログラム実行時間決定部により決定された実行時間で、前記プログラムを実行するプログラム実行部と、を備えるCPUクロック制御装置。

【請求項 2】

割り込み発生時に実行される割り込み処理に必要な処理量を検知する割り込み処理量検知部と、

割り込み発生時に発生した割り込みに対応した割り込み処理を行なう割り込み処理部と、

前記割り込み処理部が割り込み処理を行なった場合に、前記割り込み処理量検知部が検知した処理量を現在時刻に行なう処理とした上で、前記プログラム実行時間登録部によって登録された実行時間に関する要求と許容範囲を満たす範囲で、前記プログラムの実行に必用な処理量と前記割り込み処理の処理量の単位時間毎の合計が均等化するように、前記プログラムの実行時間を再決定する割り込み時実行時間調整部とを更に備え、

前記CPU動作周波数設定部は、前記プログラム実行時間決定部に均等化された処理量、または前記割り込み時実行時間調整部で均等化された処理量のいずれか新しい処理量に基づき、前記CPUの動作周波数を決定し、

前記プログラム実行部は、前記プログラム実行時間決定部により決定された実行時間、または前記割り込み時実行時間調整部で再決定された実行時間のいずれか新しい実行時間で、前記プログラムを実行することを特徴とする請求項 1 記載のCPUクロック制御装置。

【請求項 3】

動作周波数を変更できるCPUを備え、マルチタスク処理を行なう情報処理装置において、

タスクの実行時間に関する要求と、前記タスクの実行時間に関する要求の許容範囲を登録することの出来るタスク実行時間登録部と、

前記タスクの実行に必要な処理量を検知するタスク処理量検知部と、

前記タスク実行時間登録部によって登録された前記実行時間に関する要求と前記許容範囲を満たす範囲で、前記タスク処理量検知部が検知した前記タスク実行に必要な処理量の単位時間毎の合計が均等化するように、前記タスクをスケジューリングするタスクスケジューリング部と、

前記タスクスケジューリング部により均等化された前記タスクの単位時間辺りの必要な処理量の合計に基づき、前記CPUの動作周波数を決定するCPU動作周波数設定部と、

前記タスクスケジューリング部によるスケジューリングで、前記タスクを実行するタスク実行部と、を備えるCPUクロック制御装置。

【請求項 4】

動作周波数を変更できるCPUを備え、マルチタスク処理を行なう情報処理装置において、

タスクの実行時間に関する要求と前記実行時間に関する要求の許容範囲を登録することの出来るタスク実行時間登録部と、

前記タスク実行時間登録部に登録されたタスクの実行時間に関する要求と要求の許容範囲を満たすように、前記タスクを制御するタイマーイベントを決定するタスク実行時間要求解釈部と、

前記タスク実行時間要求解釈部が決定したタスクを制御するタイマーイベントを含むタイマーイベントの実行時間に関する要求と、前記タイマーイベントの実行時間の許容範囲を登録することの出来るタイマーイベント登録部と、

前記タスクの実行に必要な処理量を検知するタスク処理量検知部と、

前記タイマーイベントの実行に必要な処理量を検知するタイマーイベント処理量検知部と、

前記タイマーイベント登録部によって登録された前記タイマーイベントの実行時間に関する要求と許容範囲を満たす範囲で、タイマーイベントにより実行に移されるタスクに関し前記タスク処理量検知部が検知した必要処理量と、前記タイマーイベント処理量検知部が検知したタイマーイベントの処理量との単位時間毎の合計が均等化するように、前記タイマーイベントの実行時間を決定するタイマーイベント決定部と、

前記タイマーイベント決定部により均等化された前記プログラムの単位時間辺りの必要な処理量の合計に基づき、前記CPUの動作周波数を決定するCPU動作周波数設定部と、

前記タイマーイベント決定部が決定した実行時間でタイマーイベントを実行するタイマーイベント実行部と、を備えるCPUクロック制御装置。

【請求項5】

実行するプログラムの実行時間に関する要求と、前記プログラムの実行時間に関する要求の許容範囲を登録するステップと、

前記プログラムの実行に必要な処理量を検知するステップと、

登録された実行時間に関する要求と許容範囲を満たす範囲で、検知したプログラムの実行に必要な処理量の単位時間毎の合計が均等化するように、前記プログラムの実行時間を決定するステップと、

均等化されたプログラムの単位時間辺りの必要な処理量の合計に基づき、CPUの動作周波数を決定するステップを含み、

決定された実行時間で前記プログラムを実行することを特徴とするCPUクロック制御方法。

【書類名】 明細書

【発明の名称】 C P Uクロック制御装置及びC P Uクロック制御方法

【技術分野】

【0001】

本発明は、情報処理装置におけるC P Uの動作周波数を制御して消費電力の削減を行なうC P Uクロック制御装置及びC P Uクロック制御方法に関するものである。

【背景技術】

【0002】

情報処理装置は中央演算装置（C P U）が命令を解釈、実行することで動作するが、C P Uが消費する電力はその動作周波数（単位時間あたりに入力するクロック数）により変化する。動作周波数の制御によりC P Uの消費電力を削減する方法は、様々なものが考案、実用化されている。

【0003】

代表的な電力削減方法に、C P Uの最大動作周波数で要求される処理をなるべく早く行い、処理の必要の無い時間（アイドル時間）ではクロックを停止する方法がある。例えば、最大動作周波数が1 0 0 M H zのC P Uで5 0 Mクロック分の処理を1秒以内に行う必要が有る場合、まず1 0 0 M H zの動作周波数で0. 5秒間C P Uを駆動して処理を完了させ、残りの0. 5秒間はクロックを完全に停止するといった処理を行なう。また別の方法として、要求される時間内で処理を完了できる最小の動作周波数を算出し、その動作周波数でC P Uを駆動する方法がある。例えば、5 0 Mクロック分の処理を1秒以内に行う必要が有る場合、5 0 M H zの動作周波数で1秒間C P Uを駆動する。この2つの方法は、いずれも処理に必要な最小限のクロックだけを使うことで無駄なクロックを削減し、C P Uの消費電力を抑えるものである。

【0004】

要求される時間内で処理を完了できる最小動作周波数を算出して用いる技術として、タスク毎に要求するC P Uの性能情報を用いて、必要最小限の動作周波数でC P Uを動作させるクロック制御装置が開示されている（特許文献1参照）。また、C P Uを複数備えたシステムにおいて、タスクを遅延なく処理するように動作周波数を変化させる演算処理制御システムが発明されている（特許文献2参照）。

【0005】

1クロックの入力でC P Uが消費する電力は、その電源電圧の2乗に比例することが知られている。また、C P Uの動作周波数を上げる為には、同時に電源電圧を上げる必要があることが多い。従って、このようなC P Uでは、1クロックあたりに消費する電力が動作周波数によって異なるため、同じだけの処理量でも合計消費電力が異なる。例えば、動作周波数に比例した電源電圧が必要となるC P Uを用いると、前記従来の2つの電力削減方法では1秒間の総クロック数（＝処理量）は同じだが、この1秒間の消費電力は後者が前者1／2になる。また、同様に、同じ量の処理を同じ時間内に行う場合では、なるべく動作周波数は変えずに一定の動作周波数でC P Uを駆動した方が消費電力は少なくなる。例えば0. 5秒間2 0 0 M H z、0. 5秒間1 0 0 M H zでの駆動より、1 5 0 M H zで1秒間駆動する方が消費電力は少なくなる。電源電圧と動作周波数の関係はC P Uの設計によるが、前述した様に、処理量の要求を満たす範囲でなるべくアイドルの期間を作らず、同一の動作動作周波数でC P Uを駆動すると大きな電力削減効果を得られる場合が多い。

【0006】

また、電力の供給に電池を用いる場合、単位時間辺りの電力消費が安定している方が電池内のエネルギーを有効に利用できることが報告されており、これを利用したタスクスケジューリングアルゴリズムが開示されている（非特許文献1参照）。

【特許文献1】 特開平8－7 6 8 7 4号公報

【特許文献2】 特開2 0 0 2－9 9 4 3 3号公報

【非特許文献1】 電子情報通信学会論文誌 D－I V o l . J 8 3－D－I N o

【発明の開示】

【発明が解決しようとする課題】

【0007】

しかしながら、情報処理装置で処理を行う場合、処理の開始時間や終了の時間、あるいは実行の周期といった実行時間が指定される場合が多い。例えば、映像の描画を行う場合、映像1フレームに相当する時間毎に周期的に処理を行う必要がある。状況によっては、周期の異なる複数の周期的処理や、周期的ではないが実行する時間を指定された処理を多数同時に実行することもある。このような時間を指定された処理は、例えば、情報処理装置のOSが管理するタイマーイベントを用いたり、あるいはタイマーイベントからタスクを起動する形で実現することができる。

【0008】

前記従来の技術で述べたようなCPUでは、動作周波数を制御して、なるべくアイドルの期間を作らず同一の動作周波数で動作させると、大きな電力削減効果を得られる。しかしながら、一般に、時間を指定された処理、例えばタイマーイベントやそれによって起動されるタスクは、特定の時間に集中することがある。また、逆に特定の時間は処理が存在せずアイドル期間になることがある。つまり、CPUに要求される処理量の変動するので、前記特許文献1のような発明を用いても、CPUの動作周波数はその時間に要求される処理量に合わせ変動することになり、動作周波数が変動するため、電力削減効果は減少するという課題を有していた。

【0009】

一方、タイマーイベントなどの実行時間を指定する処理において、その要求の厳密さは様々である。例えば、実行開始時間を指定している場合でも、一定の範囲内なら実行開始時間が遅れても許容できるものもある。しかし、このような実行時間要求の許容範囲を利用して、CPUが行なうべき処理量を平滑化し、CPU動作周波数の変動を抑える制御は、前記特許文献1、特許文献2、非特許文献1に記載の発明や技術では不可能であった。

【0010】

本発明は、上記課題を鑑みて成されたものであり、情報処理装置においてCPUに要求される処理量を平滑化することによりCPUの消費電力の削減を行うものである。

【課題を解決するための手段】

【0011】

本発明のCPUクロック制御装置は、動作周波数を変更できるCPUを備えた情報処理装置において動作し、実行するプログラムの実行時間に関する要求とその要求の許容範囲を登録することの出来るプログラム実行時間登録部と、プログラムの実行に必要な処理量を検知するプログラム処理量検知部と、登録された実行時間に関する要求と許容範囲を満たす範囲で、プログラムの実行に必要な処理量の単位時間毎の合計が均等化するように、プログラムの実行時間を決定するプログラム実行時間決定部を備える。これにより、プログラムの実行時間に関する要求の許容範囲内で、必要とする処理量を均等化するプログラムの実行時間を決定することが出来る。

【0012】

更に、本発明のCPUクロック制御装置は、均等化されたプログラムの単位時間辺りの必要な処理量の合計に基づき、CPUの動作周波数を決定するCPU動作周波数設定部と、決定された実行時間でプログラムを実行するプログラム実行部とを備える。これによりCPUの動作周波数の変動を極力抑える制御を行い、課題を解決する。

【発明の効果】

【0013】

本発明によれば、要求される処理の実行時間に関する要求を満たしながら、一層のCPU消費電力の削減を行い、電力の有効な活用を行なうことが可能である。

【発明を実施するための最良の形態】

【0014】

以下本発明の実施の形態について、図面を参照しながら説明する。

【0015】

（実施の形態1）

図1は、本発明の実施の形態1のCPU動作周波数制御を行なう情報処理装置を示す図である。情報処理装置は少なくとも一つ以上の中央演算装置（CPU）1とメモリ装置2を備えている。また、図示しないが、情報処理装置は入力装置、出力装置のような他の機器を備えていてもよい。入力装置、出力装置については、他の実施の形態でも同様である。

【0016】

CPU1は、動作周波数を変更可能であり、その変更はオペレーティングシステム（OS）100、より正確には後述するOS100内のCPU動作周波数設定部104により行われる。

【0017】

メモリ装置2には、少なくとも一つ以上のプログラム10、情報処理装置のOS100が格納されている。メモリ装置2は、十分な機能及び容量があればランダムアクセスメモリ（RAM）、フラッシュメモリのような任意のタイプのものを使用することができる。また、メモリ装置2は単一のメモリ装置で構成されている必要はなく、同種の複数のメモリ装置又はリードオンリーメモリ（ROM）を含む異なる種類のメモリ装置の組み合わせであってもよい。さらに、メモリ装置2以外に、例えばハードディスクのような外部記憶装置を設け、情報処理装置の動作に問題の無い範囲で外部記憶装置にメモリ装置の内容を移動させても良い。

【0018】

本実施の形態でのプログラム10とは、情報処理装置で実行される各々の処理を情報処理装置のプログラムとして記述したものである。本実施の形態では、OS100の制御の元でプログラム10を実行することで、情報処理装置は各種の処理を行う。

【0019】

OS100は、本実施の形態のCPUクロック制御を行なう為、プログラム実行時間登録部101、プログラム処理量検知部102、プログラム実行時間決定部103、CPU動作周波数設定部104、プログラム実行部105を備えている。

【0020】

プログラム実行時間登録部101は、プログラムの実行時間に関する要求とその許容範囲についての情報を登録する。プログラム実行時間登録部101は、登録されたプログラムの時間に関する要求とその許容範囲をプログラム実行時間決定部103に通知する。

【0021】

プログラムの実行時間に関する要求とは、例えば、開始時間に関する要求、つまりいつプログラムの実行を開始するか要求である。あるいは、いつまでに処理を終える必要があるかといった終了時間に関する要求や、周期的にプログラムを実行する場合の実行周期等である。どのような要求を登録可能にするかは、情報処理装置の使用の目的や環境に合わせて設計することが出来る。また、許容範囲とは、実行時間に関する要求に対してどの程度の変動が許されるかの情報である。例えば、開始時間に関する要求に対しては、許される開始時間の遅れである。1秒後の実行開始を要求するが、0.1秒の開始時刻の遅れが許される場合は、実行時間に関する要求が1秒後、許容範囲が0.1秒となる。10秒ごとの周期実行を要求し1秒の周期の変動が許容できる場合は、実行時間に関する要求が10秒周期、許容範囲は1秒となる。どのような許容範囲を登録可能にするか、あるいはその表記の方法は、実行時間に関する要求と同様に、情報処理装置の使用の目的や環境に合わせて設計することが出来る。

【0022】

登録される情報である実行時間に関する要求と許容範囲は、例えば、プログラムを情報処理装置に登録またはインストールする時に、実行時間要求を登録する関数を呼び出して引数として渡せば良い。登録やインストールを行なう専用のプログラムが有るのが普通の

ため、登録やインストールを行なうプログラムが、前述した実行時間要求を登録する関数を出すことで実現できる。

【0023】

プログラム処理量検知部102は、プログラム10の実行時に必要とする処理量を検知し、プログラム実行時間決定部103に通知する。処理量の検知は、例えば、各プログラム内に必要とする処理量を記述しておき、実行前にプログラム10から関数の引数として渡すことで可能である。あるいはプログラムのヘッダ等に処理量を記述しておき、プログラム処理量検知部102がその値を読み出しても良い。あるいは、複数回実行されるプログラムでは、プログラムを実際に実行して終了までかかった時間を記録し、その時間でCPUの動作周波数を積分（CPU動作周波数が一定の場合は乗算）して要した処理量を実測しても良い。処理量の検知方法は、実際に検知が可能であれば任意の方法を用いることが出来る。また、本発明ではプログラムの必要とする処理量は正確な値である必要は必ずしもなく、処理量が大、中、小のような、近似したものでも効果を得ることが出来る。これは他の実施の形態でも同様である。

【0024】

プログラム実行時間決定部103は、通知されたプログラム10の実行時間に関する要求と許容範囲、必要な処理量に基づき、プログラム10の実行の時間を決定する。ここで、プログラム実行時間決定部103は、実行時間に関する要求とその許容範囲を満たす範囲で各プログラム10の実行時間を決定する。同時に、予め定められた単位時間毎で、実行されるプログラムの必要処理量の合計がなるべく一定になるように実行の時間を決定する。つまり、単位時間毎のプログラムの必要処理量が極力等しくなるようにプログラムの実行時間を決定する。

【0025】

CPU動作周波数設定部104は、プログラム実行時間決定部103により平滑化された処理量に基づいてCPUの動作周波数を決定し、設定する。設定される周波数は、例えば、必要とされる処理を各単位時間で丁度終了し、アイドル時間を作らない最小の動作周波数に設定する。単位時間が1秒間で、ある単位時間の間（ここでは1秒間）に100Mクロック分の処理が必要とされる場合、この1秒間は100MHzの動作周波数に設定する。あるいは、実際の処理量に余裕を持たす為、最低限必要な動作周波数に一定量を加えた動作周波数に設定しても良い。単位時間毎に行なわなければならない処理量は、プログラム実行時間決定部103により平滑化されているので、CPU動作周波数設定部104により設定される動作周波数の変動は、平滑化を行わない場合に比べて少なくなる。前述した様に、同じだけの処理量でもCPUの動作周波数の変動は少ない方が消費電力は少なくなる。従って、平滑化を行わない場合に比べ、CPUの消費する電力は少なくなる。

【0026】

プログラム実行部105は、プログラム実行時間決定部103の決定した実行時間でプログラム10を実行する。

【0027】

図2は、本実施の形態のCPUクロック制御装置の動作を説明するフローチャートである。

【0028】

ステップS2-1からステップS2-5までは、プログラム実行時間登録部101とプログラム処理量検知部102の動作、ステップS2-6からステップS2-9までは、プログラム実行時間決定部103、CPU動作周波数設定部104、プログラム実行部105の動作である。

【0029】

ステップS2-1で情報処理装置が実行するプログラムが登録される。ステップS2-2で登録されたプログラムの実行時間に関する要求と許容範囲が登録される。ステップS2-3では、登録された実行時間に関する要求と許容範囲をプログラム実行時間決定部1

03に通知する。ステップS2-4ではプログラムの必要な処理量の検知が行なわれる。前述の通り、ステップS2-4は、例えばプログラム10の方から関数を呼び出して必要処理量を渡したり、実際のプログラムの実行を待って必要処理量の実測をする。ステップS2-5では、検知されたプログラムの必要処理量をプログラム実行時間決定部103に通知する。なお、図2のフローチャートでは、時間に関する要求と許容範囲の登録と通知(S2-2、S2-3)を、必要処理量の検知と通知(S2-4、S2-5)より先に行なっているが、必ずしもその必要は無い。これらは独立したものなので、任意の順番で実行が可能である。

【0030】

ステップS2-6は、プログラム実行時間決定部103の呼び出しである。プログラム実行時間決定部103の呼び出しは、新しいプログラムが登録された直後、つまりステップS2-5の直後で良い。あるいは、特定の周期で周期的に行ったり、一つのプログラムが処理を完了した時に呼び出して良い。

【0031】

S2-7は、通知されている全てのプログラムの実行時間の決定を行なう。前述の通り、ステップS2-3で通知されている各プログラムの実行時間要求と許容範囲を満たす範囲で、ステップS2-5で通知された各プログラムの必要処理量の単位時間辺りの合計が、極力均等化するように実行時間を決定する。

【0032】

ステップS2-7の実行時間の決定は、制約を満たしながらプログラムの実行時間を配置する一般的な問題であり、様々なアルゴリズムを用いることが出来る。例えば、対象となるプログラム10が比較的少数なら、考えられる全ての実行時間のパターンを列挙して処理量の変動を比較しても良い。ただし、本発明では完全な均等化を行なう必要は必ずしも無い。CPU動作周波数の変動を少なくした分だけ消費電力を少なく出来るので、完全でなくても均等化した分だけの消費電力削減効果を得られる。従って、例えば、最も必要処理量の大きくなる時間から、一つのプログラムの実行を別の時間へ移動させるだけでも効果がある。また、プログラムの必要処理量が正確な値でなく近似値の場合でも、ある程度の均等化は可能なので、効果を得ることが出来る。また、ここでは全てのプログラムの実行時間を決定する必要はなく、現在時刻から一定先の時刻まででも良い。均等化には計算時間が必要となるので、どの程度正確な均等化を行なうかは、情報処理装置の使用の目的や環境に合わせて設計すれば良い。

【0033】

ステップS2-8は、CPUの動作周波数を決定し設定する。S2-9はプログラムの実行である。プログラムは決定された時間に実行されるので、実際は、ステップS2-9はステップS2-8終了後直ちに実行されるわけではなく、決定された実行時間を迎え時に実行される。

【0034】

図3は、本実施の形態での実際の処理の一例である。

【0035】

図3(a1)は、プログラムが要求している実行時間そのままプログラムを配置した場合を表している。(a1)は、図2のステップS2-1からステップS2-5の結果、プログラム実行時間決定部103に通知される情報であり、横軸は時刻、縦軸はその時間に必要される処理量である。四角のブロックはプログラムを表しており、ブロックの縦の大きさはそれぞれのプログラムの必要処理量に対応する。(a1)では、AからDまでの4つのプログラムが登録されている。プログラム名であるアルファベットの後の値はプログラムの必要処理量(単位はMクロック)であり、その後ろの値は実行時間要求と許容範囲である。プログラムAの場合、必要処理量が100、実行時間要求が0、許容範囲が+2となる。

【0036】

図3の例では、実行時間の要求は、その値の時間からその次の時間までの間隔での実効

と処理の完了を要求しているとする。Aの場合は、時刻0から1の間での実行と処理の完了を要求していることになる。以下では、これを「時刻0での実行を要求している」という様に表記する。Cの場合は、時刻2から3の間での実行と処理完了を要求しているので、「時刻2での実行を要求している」となる。

【0037】

許容範囲の値は、実行時間の変動の範囲を表しており、+1の場合は時間1の実行の遅れまで許容、+2なら時間2の遅れまで許容となる。プログラムAの場合は許容範囲が+2なので、要求時刻0での実行を要求しているが時刻1や2での実行でも許容できる。Bは時刻0での実行を要求しており、他の時間での実行は出来ない。C、Dは時刻2での実行を要求しており、この時刻で実行しなければならない。

【0038】

図3(b1)は、(a1)の実行時間でプログラムAからDを実行し、各時間でそれぞれの処理を必要最小限の動作周波数で完了する、つまりアイドル期間をつくらないようにCPU動作周波数を設定した場合を表している。

【0039】

図3(a2)は、ステップS2-6、S2-7が実行されて決定されたプログラムの実際の実行時間を表している。ステップS2-7の結果、各時間毎の必要周波数が均等化するようにプログラムAを時刻1で実行することが決定されている。なお、ここでは計算に必要な処理量を考えて、現在時刻より3時刻先までのプログラム実行のみを均等化している。つまり図中の範囲のみで考えている。

【0040】

図3(b2)は(a2)のプログラム実行でのCPU動作周波数の設定である。(b1)と比較すると動作周波数の変動が少なくなっており、この期間でのCPUの消費電力が削減される。また、各プログラムはそれぞれの許容範囲を満たして実行されている。

【0041】

図4は、本実施の形態の実際の処理の別の例である。図4(a1)は、プログラムの実行時間の要求である。プログラムAは周期2での周期的実行を要求しており、図4(a1)ではこれを実行時間要求がP2と表記している。周期的実行なので、プログラムAは時刻0、2、4での実行を要求している。ただしプログラムAは、実行される時間が1遅れても許容できる。(a2)はプログラムの実行時間の決定後であり、この例では、時刻2での実行を要求しているプログラムAは、実際には時刻3で実行される。

【0042】

(b1)、(b2)は処理に必要な最小の動作周波数を用いる場合の動作周波数である。(b2)は完全な均等化ではないが(b1)より動作周波数の変動が少なくなり、CPUの消費電力は削減される。また、各プログラムはそれぞれの許容範囲を満たして実行されている。

【0043】

以上の様に、本発明ではプログラムの実行時間に関する要求を許容範囲内で満たした上で、CPUの消費電力を削減することが可能である。なお、本実施の形態でのプログラム10は、情報処理装置で実行される全ての処理である必要はなく、その内の特定の一部でも良い。同様に、全てのプログラム10が、時間要求と許容範囲を登録する必要は無い。処理量を平滑化されるのが全ての処理の一部であっても、その分の処理量は平滑化されるので、消費電力の削減効果を得ることが可能である。

【0044】

なお、本発明の実施の形態ではOS内に、プログラム実行時間登録部101、プログラム処理量検知部102、プログラム実行時間決定部103、CPU動作周波数設定部104、プログラム実行部105設けたが、OS外部に設けることも可能である。

【0045】

(実施の形態2)

図5は、本発明の実施の形態2のCPU動作周波数制御を行なう情報処理装置を示す図

である。CPU 1、メモリ装置 2 は実施の形態 1 と同じである。また、プログラム 20 は実施の形態 1 のプログラム 10 と同じである。CPU 1 の動作周波数の変更、及びプログラム 20 の制御は OS 200 内の CPU 動作周波数設定部 207 により行われる。

【0046】

OS 200 は、本実施の形態の CPU クロック制御を行なう為、プログラム実行時間登録部 201、プログラム処理量検知部 202、プログラム実行時間決定部 203、割り込み処理量検知部 204、割り込み処理部 205、割り込み時実行時間調整部 206、CPU 動作周波数設定部 207、プログラム実行部 208 を備えている。

【0047】

プログラム実行時間登録部 201 は実施の形態 1 のプログラム実行時間登録部 101 と同様であり、プログラムよりその時間に対する要求と、許容範囲についての情報を受け取り、プログラム実行時間決定部 203 と割り込み時実行時間調整部 206 に通知する。同様に、プログラム処理量検知部 202 は実施の形態 1 のプログラム処理量検知部 102 と同じであり、プログラムの必要な処理量をプログラム実行時間決定部 203 と割り込み時実行時間調整部 206 に通知する。

【0048】

プログラム実行時間決定部 203 は、実施の形態 1 のプログラム実行時間調整部 103 と同様に、プログラムの実行時間に関する要求、許容範囲、必要な処理量に基づき、一定時間毎のプログラムの必要処理量の合計がなるべく均一になるようにプログラムの実行の時間を決定する。

【0049】

割り込み処理量検知部 204 は、割り込み発生時に実行される処理（割り込みハンドラ）が必要とする処理量を、割り込み時実行時間調整部 206 に通知する。割り込み処理が必要とする処理量の検知は、例えば、予め処理量を見積もっておき、割り込みハンドラを OS 200 に登録する時にその値を渡せば可能である。あるいは、実際に割り込み処理が行われた際に要した処理量を実測して、それに基づき検知しても良い。

【0050】

割り込み処理部 205 は、割り込み信号が発生した際に対応した割り込み処理を実行し、合わせて割り込み処理が発生したことを割り込み時実行時間調整部 206 に通知する。

【0051】

割り込み時実行時間調整部 206 は、プログラム実行時間決定部 203 と同様に、プログラムの実行時間に関する要求、許容範囲、必要な処理量に基づき、一定時間毎のプログラムの必要処理量の合計がなるべく一定になるようにプログラムの実行の時間を決定する。但し、割り込み時実行時間調整部 206 は、割り込み処理部 205 より割り込み発生が通知された場合にのみプログラム実行時間の決定を行なう。また、プログラム実行時間の決定は、割り込み処理量検知部より通知されている割り込み処理の処理量を、割り込み発生時間に行う処理として追加した上で、単位時間毎の処理量を極力均一化するようにプログラムの実行の時間を決定する。つまり、割り込み処理に費やした処理量を計算に入れた上で処理量が平滑化するようにプログラムの実行時間を再決定する。

【0052】

CPU 動作周波数設定部 207 は、実施の形態 1 の CPU 動作周波数設定部 104 と同様に、プログラム実行時間決定部 203 により平滑化された単位時間毎に必要なになる処理量に基づいて CPU の動作周波数を決定し、設定する。さらに、割り込み時実行時間調整部 206 がプログラム実行時間を再決定した場合は、割り込み時実行時間調整部 206 により平滑化された処理量に基づいて CPU 動作周波数を決定する。つまり、最新の平滑化の結果に基づき CPU 動作周波数を設定する。CPU 動作周波数設定部 207 により設定される動作周波数の変動は平滑化を行わない場合に比べて少なくなり、CPU の消費する電力は削減される。また、割り込み処理が実行された場合は、それを加えた上で平滑化された必要処理量に基づいて CPU 動作周波数が決定されるので、割り込みが発生した場合でも適切な CPU 動作周波数を用いることが出来る。

【 0 0 5 3 】

プログラム実行部 2 0 8 は、プログラム実行時間決定部 2 0 3 の決定した実行時間でプログラム 1 0 を実行するが、割り込み時実行時間調整部 2 0 6 がプログラム実行時間を再決定した場合は、再決定された実行時間でプログラム 1 0 を実行する。つまり、割り込み発生も含めた最新の決定に基づきプログラム 1 0 を実行する。

【 0 0 5 4 】

図 6 は本実施の形態の C P U クロック制御装置の動作を説明するフローチャートである。

【 0 0 5 5 】

ステップ S 6 - 1 からステップ S 6 - 5 は、プログラムの時間要求と許容範囲の登録、及び処理量の検知であり、実施の形態 1 のステップ S 2 - 1 からステップ S 2 - 5 (図 2) と同様である。

【 0 0 5 6 】

ステップ S 6 - 6 はプログラム実行時間決定部 2 0 3 の呼び出しであり、実施の形態 1 のステップ S 2 - 6 と同様である。ステップ S 6 - 7 は、ステップ S 2 - 7 と同様である。ステップ S 6 - 8 では、決定した実行時間と必要な処理量を C P U 動作周波数設定部 2 0 7 とプログラム実行部 2 0 8 に通知する。

【 0 0 5 7 】

ステップ S 6 - 9 からステップ S 6 - 1 1 は割り込み処理部 2 0 5 、割り込み時実行時間調節部 2 0 6 の処理の流れである。ここでは、割り込み処理の処理量は、事前に検知されているとし図示していない。ステップ S 6 - 9 は割り込みの発生であり、ステップ S 6 - 1 0 、 S 6 - 1 1 は割り込みを契機に行なわれる。ステップ S 6 - 1 0 は、割り込み処理に必要な処理量を含めた上でのプログラム実行時間の決定である。割り込み処理を、現在の時間に必要な処理とみなす以外は、ステップ S 6 - 7 と同様である。ステップ S 6 - 1 1 では、決定した実行時間と必要な処理量を C P U 動作周波数設定部 2 0 7 とプログラム実行部 2 0 8 に通知する。

【 0 0 5 8 】

ステップ S 6 - 1 2 からステップ S 6 - 1 4 は、C P U 動作周波数 2 0 7 とプログラム実行部 2 0 8 の処理の流れである。ステップ S 6 - 1 2 は、ステップ S 6 - 8 または S 6 - 1 1 によるプログラム実行時間と必要処理量の通知であり、ステップ S 6 - 1 3 、 S 6 - 1 4 はこれを契機に実行される。S 6 - 1 3 は、通知された必要処理量に基づく C P U 動作周波数の決定である。ステップ S 6 - 1 4 は、決定された実行時間でのプログラム 2 0 の実行である。

【 0 0 5 9 】

図 7 は、本実施の形態の実際の処理の一例である。図 7 (a 1) は、図 6 のステップ S 6 - 1 から S 6 - 8 の結果、プログラムの実行時間が決定され、その後に、割り込みが発生し必要処理量 1 5 0 の割り込み処理が発生した状態を示している。割り込み発生により、ステップ S 6 - 9 、 S 6 - 1 0 が実行されプログラムの実行時間が再決定される。この例では、プログラム A 、 B の実行の時間が変更され (a 2) となる。必要な処理量が均等化されて動作周波数の変動が少なくなり、C P U の消費電力は削減される。また、各プログラムはそれぞれの許容範囲を満たして実行されている。

【 0 0 6 0 】

なお、本実施の形態においても、実施の形態 1 と同様に、プログラム 2 0 は情報処理装置で実行される全ての処理である必要はなく、その内の特定の一部であってもよい。また、時間要求と許容範囲を登録するのは一部のプログラムでも良い。

【 0 0 6 1 】

なお、本発明の実施の形態では O S 内に、プログラム実行時間登録部 2 0 1 、プログラム処理量検知部 2 0 2 、プログラム実行時間決定部 2 0 3 、割り込み処理量検知部 2 0 4 、割り込み処理部 2 0 5 、割り込み時実行時間調整部 2 0 6 、C P U 動作周波数設定部 2 0 7 、プログラム実行部 2 0 8 を設けたが、O S 外部に設けることも可能である。

【0062】

(実施の形態3)

図8は、本発明の実施の形態3のCPU動作周波数制御を行なう情報処理装置を示す図である。CPU1とメモリ装置2は実施の形態1と同じである。

【0063】

タスク30は、情報処理装置が処理を行う単位であり、OS300により時分割的に実行される。このような処理形態はマルチタスク処理と呼ばれ、一般的に用いられる処理形態である。本実施の形態の情報処理装置はマルチタスク処理を行う。

【0064】

OS300は、本実施の形態のCPUクロック制御を行なう為、タスク実行時間登録部301、タスク処理量検知部302、タスクスケジューリング部303、CPU動作周波数設定部304、タスク実行部305を備えている。

【0065】

タスク実行時間登録部301は、タスクの実行時間に対する要求と、その許容範囲についての情報を登録し、その情報をタスクスケジューリング部303に通知する。タスクの実行時間に関する要求とは、例えば、タスクの開始時間に関する要求、終了時間に関する要求、実行周期等である。どのような要求を登録可能にするかは、OS300のタスクスケジューリングの方法に合わせて設計することが出来る。また、同様に、どのような許容範囲を登録可能にするか、あるいはその表記の方法は、OS300のタスクスケジューリングの方法に合わせて設計される。

【0066】

タスク処理量検知部302は、各タスクの必要とする処理量を検知し、タスクスケジューリング部303に通知する。処理量の検知は、例えば、タスクのスケジューリングの方法を指定する際に、予め見積もられた処理量を合わせてOS300に渡すことで可能である。実際にタスクを実行して要した処理量を求めて、それに基づき決定しても良い。

【0067】

一般的にマルチタスクの情報処理装置は、タスクのスケジューリングの方法を指定するインターフェイスを備えていることが多い。例えば、周期的なタスクの実行を関数呼び出しで要求し、関数の引数で周期を指定するインターフェイスである。特定のタスクに関し、特定の時間毎にかならず一定量の処理を行なうといった指定などもある。そのようなインターフェイスが有るなら、その指定を実行時間に関する要求や処理量の検知に用いることが出来る。

【0068】

タスクスケジューリング部303はタスクのスケジューリングを行なう。タスクスケジューリング部303は、時間に関する要求とその許容範囲を満たす範囲でタスクをスケジューリングするが、同時に一定時間毎のタスクの必要処理量の合計がなるべく一定になるようにタスクをスケジューリングする。

【0069】

CPU動作周波数設定部304は、タスクスケジューリング部303により平滑化された単位時間毎に必要な処理量に基づいてCPUの動作周波数を決定し、設定する。タスク30が単位時間毎に要求される処理量は平滑化されているので、CPU動作周波数設定部304により設定される動作周波数の変動は平滑化を行わない場合に比べて少なくなり、CPU1の消費する電力は削減される。

【0070】

タスク実行部305は、タスクスケジューリング部303の行なったスケジューリングに基づきタスク30を実行する。

【0071】

図9は本実施の形態での処理の流れを示すフローチャートである。

【0072】

ステップS9-1で情報処理装置にタスクが登録される。ステップS9-2で登録され

たプログラムの実行時間に関する要求と許容範囲が登録される。ステップS 9－3では、登録された実行時間に関する要求と許容範囲がタスクスケジューリング部3 0 3に通知される。ステップS 9－4ではタスクの必要な処理量の検知が行なわれる。前述の通り、ステップS 9－4は、タスクから関数を呼び出して必要処理量を渡したり、実際に必要処理量を実測することで可能である。ステップS 9－5では、タスクの必要な処理量をタスクスケジューリング部3 0 3に通知する。なお、図9のフローチャートでは、時間に関する要求と許容範囲の登録と通知（S 9－2、S 9－3）を、必要処理量の検知と通知（S 9－4、S 9－5）より先に行なっているが、これらは独立したもののなので任意の順番で実行が可能である。

【0 0 7 3】

ステップS 9－6は、タスクスケジューリング部3 0 3の呼び出しである。タスクスケジューリング部3 0 3の呼び出しは、通常のタスクスケジューリングのタイミングで行なえば良い。ステップS 9－7は、タスクの実行時間の決定を行なう。ここでは各タスクの時間要求と許容範囲を満たす範囲で、各タスクの必要処理量の単位時間辺りの合計が、極力均等化するようにスケジューリングする。ステップS 9－8では、CPUの動作周波数を決定し、設定する。ステップS 9－9では、ステップS 9－7で行なわれたタスクスケジューリングに基づきタスクが実行される。

【0 0 7 4】

本実施の形態での実際の処理の例としては、実施の形態1の例で示したもの（図3、図4）と同様のものがある。実行時間を指定されたタスクのスケジューリングを図3、図4と同様に行なうことが可能である。特に、図4と同様なスケジューリングは、特定のタスク（図4ではプログラムA）を周期的に実行するスケジューリングであるが、同時に一定周期毎（図4では2）に必ず定められた量の処理量（図4では5 0）を行なうようなスケジューリングでもある。

【0 0 7 5】

図1 0は本実施の形態での実際の処理の別の例である。図1 0の例では、タスクのデットラインスケジュールを行なう。デットラインスケジュールは、タスクの処理が完了しなければならない時刻を指定し、その時刻（デットライン）が早いタスクから実行する。

【0 0 7 6】

図1 0（a 1）は、要求しているデットラインが早いものから順にタスクを実行する通常のスケジューリングである。四角のブロックはタスクを表しており、（a 1）では、AとBの2つタスクがスケジューリングの対象である。タスク名であるアルファベットの後の値はタスクの必要処理量であり、その次の値は実行時間要求と許容範囲である。この例では、実行時間の要求はデットラインであり、デットラインであることをアルファベットのEで表す。許容範囲はデットラインの許容範囲、つまりデットラインからどの位の遅れが許容できるかである。タスクAの場合、必要処理量が2 0 0、デットラインが1、許容範囲が+ 2となる。これは、タスクAは時刻1までの終了を要求するが、最悪時刻3までの終了を認めることを意味する。（a 1）は、図9のステップS 9－1からステップS 9－5の結果、タスクスケジューリング部3 0 3に通知される情報である。

【0 0 7 7】

図1 0（b 1）は、（a 1）のスケジューリングでタスクA、Bを実行し、それぞれの処理を必要最小限の動作周波数で完了し、アイドル期間をつくらないようにした場合のCPU動作周波数を表している。

【0 0 7 8】

図1 0（a 2）は、ステップS 9－6、S 9－7で行なわれたスケジューリングの結果である。ここでは、時間毎に要求される処理量を極力平滑化するスケジューリングが行なわれている。

【0 0 7 9】

図3（b 2）は（a 2）のプログラム実行でのCPU動作周波数の設定である。（b 1）と比較すると動作周波数の変動が少なくなっており、この期間でのCPUの消費電力が

削減される。また、タスク A、B はそれぞれの許容範囲を満たして実行されている。

【0080】

このように本実施の形態では、実行時間の要求と許容範囲、必要処理量の指定により様々なスケジューリングが可能である。なお、本実施の形態でタスク 30 の全てが実行時間に関する要求を登録する必要は無い。処理量を平滑化されるのが全てのタスクの一部であっても、限定的ではあるがある程度の効果を得ることが可能である。

【0081】

また、本実施の形態においても実施の形態 2 と同様に、割り込み処理量検知部と、割り込み処理部と、割り込み発生時にタスクの再スケジューリングを行なう割り込み時実行時間調整部を設けることも可能である。この場合は、割り込み発生時に、割り込み処理の分を加えた上で処理量が均等化するようにタスクを再スケジューリングする。これにより、割り込みの発生にも対応することが出来る。

【0082】

なお、本発明の実施の形態では OS 内に、タスク実行時間登録部 301、タスク処理量検知部 302、タスクスケジューリング部 303、CPU 動作周波数設定部 304、タスク実行部 305 を設けたが、OS 外部に設けることも可能である。

【0083】

(実施の形態 4)

図 11 は、本発明の実施の形態 4 の CPU 動作周波数制御を行なう情報処理装置を示す図である。情報処理装置は少なくとも一つ以上の CPU 1 とメモリ装置 2 を備えている。CPU 1 とメモリ 2 は実施の形態 1 と同じである。タスク 40 は実施の形態 3 のタスク 30 と同じである。CPU 1 の動作周波数の変更、及びタスク 40 の制御は OS 400 により行われる。

【0084】

タイマーイベント 41 は、実行する時間を指定して OS 400 に登録される処理である。タイマーイベント 41 の登録と実行の機能は、実時間に応じた処理を行なうため、多くの OS において実現されている。タイマーイベント 41 は単体でも利用できるが、本実施の形態ではタスク 40 の制御にも用いる。

【0085】

OS 400 は、本実施の形態の CPU クロック制御を行なう為、タスク実行時間登録部 401、タスク実行時間要求解釈部 402、タスク処理量検知部 403、タイマーイベント処理量検知部 404、タイマーイベント登録部 405、タイマーイベント調整部 406、CPU 動作周波数設定部 407、タイマーイベント実行部 408、タスク実行部 409 を備えている。

【0086】

タスク実行時間登録部 401 は、実施の形態 3 のタスク実行時間登録部 301 と同様である。タスク実行時間登録部 401 は、タスクの実行時間に関する要求と許容範囲についての情報を登録し、その情報をタスク実行時間要求解釈部 402 に通知する。

【0087】

タスク実行時間要求解釈部 402 は、通知されたタスクの実行時間に関する要求を満たすように、タスク制御を行なうタイマーイベントを決定する。例えば、特定時間にタスクの実行開始が要求されている場合は、タスクを起床するタイマーイベントをその時間に実行するようにする。この様なタイマーイベントによるタスクの制御は現在一般的に行なわれているので、本実施の形態でのタスク制御のタイマーイベントもそれと同様にすれば良い。合わせて、タスク実行時間要求解釈部 402 は、通知されたタスクの実行時間に関する要求の許容範囲からタスク制御のタイマーイベントの実行時間の許容範囲を求める。例えば、タスクの実行開始時間の許容範囲として一定時間の遅れが許されるなら、その時間をタスクを起床するタイマーイベント実行開始時間の許容範囲とする。周期的タスク実行が要求されるならば、要求されている周期でタスクを起床するタイマーイベントの実行し、タスク実行周期に許容範囲があればその許容範囲をタイマーイベントの周期実行の許容

範囲とする。タスク実行時間要求解釈部402は、その実行時間、許容範囲とともに、決定したタスク制御のタイマーイベントの登録をタイマーイベント登録部405に要求する。

【0088】

タスク処理量検知部403は、実施の形態3のタスク処理量検知部302と同様である。タスク処理量検知部403は、検知したタスクの必要とする処理量をタイマーイベント決定部406に通知する。

【0089】

タイマーイベント処理量検知部404は、タイマーイベントが必要とする処理量を検知し、タイマーイベント決定部406に通知する。必要な処理量の検知は、タスク処理量検知部403と同様に、例えば、各タイマーイベント内で予め見積もった必要処理量を通知したり、実際に実行して要した処理量を実測してそれに基づき決定することで可能である。

【0090】

タイマーイベント登録部405は、タスク実行時間要求解釈部402より要求されたタイマーイベントを、タイマーイベントの実行時間に対する要求と許容範囲と共に登録する。また、タイマーイベント登録部405は、タスクとは無関係なタイマーイベントを外部からの要求に基づき登録できても良い。この場合も、タイマーイベント実行時間に対する要求、許容範囲を登録しても良い。タイマーイベント登録部405は、登録されたタイマーイベントに関する情報をタイマーイベント決定部406に通知する。

【0091】

タイマーイベント決定部406は、通知されたタイマーイベントの実行時間に関する要求と許容範囲、タイマーイベントによりタスクが実行される場合はタスクに必要な処理量、及びタイマーイベント自身が必要な処理量に基づき、タイマーイベントを実際に実行する時間を決定する。この時、タイマーイベント決定部406は、タイマーイベントの実行時間に関する要求とその許容範囲を満たす範囲で、タイマーイベントを実行する時間を決定する。また、同時にタイマーイベントを実行することで必要となる処理量、つまり、タイマーイベント自身と、タイマーイベントにより実行されるタスクの必要な処理量が、一定時間毎でなるべく均一になるようにタイマーイベント実行する時間を決定する。従って、タイマーイベントとそれによるタスク実行が一定時間毎に必要な処理量の全てが平滑化される。

【0092】

CPU動作周波数設定部407は、タイマーイベント決定部406により平滑化された処理量に基づいて、CPUの動作周波数を決定して設定する。

【0093】

タイマーイベント実行部408は、タイマーイベント決定部406が決定した実行時間でタイマーイベント41を実行する。

【0094】

タスク実行部409は、タスク40を実行する。

【0095】

図12は本実施の形態の処理の流れを表すフローチャートである。ステップS12-1からステップS12-5は、タスク実行時間登録部401、タスク実行時間要求解釈部402、タスク処理量検知部403の処理の流れを示している。ステップS12-6からステップS12-7までは、タイマーイベント登録部405、タイマーイベント決定部406、CPU動作周波数設定部407、タイマーイベント実行部408の処理の流れを示している。

【0096】

ステップS12-1で情報処理装置にタスクが登録される。ステップS12-2では、登録されたタスクの実行時間に関する要求と許容範囲が登録される。ステップS12-3では、登録されたタスクの実行時間に関する要求と許容範囲を満たすようにタスクを制御

するタイマーイベントと、タイマーイベントの実行時間に関する要求、許容範囲が決定される。例えば、タスクが1秒後の起床を要求し、許容できる起床時間の遅れを0.1秒としている場合、タスクを起床するタイマーイベントを1秒後に0.1秒の遅れの範囲で実行するように決定する。ステップS12-3では、タスクの実行に必要な処理量が検知される。ステップS12-3は必ずしもこの時点ある必要はなく、タスクの必要処理量の情報が必要となる前なら、任意の時点で良い。ステップS12-5は、ステップS12-3で決定されたタイマーイベントと、実行時間要求、許容範囲の登録をタイマーイベント登録部403に要求する。

【0097】

ステップS12-6は、タイマーイベントの登録要求であり、ステップS12-7からステップS12-13はタイマーイベントの登録要求を契機に実行される。ステップS12-7では登録要求されているタイマーイベントの処理量の検知が行なわれる。ステップS12-7は必ずしもこの時点で行なう必要は無く、タイマーイベントの処理量の情報が必要になる前ならどの時点でも良い。

【0098】

ステップ12-8では、登録が要求されているタイマーイベントがタスクを起床するものかどうか判定される。タスクを起床する場合はステップ12-9で、タイマーイベント自身の必要処理量に起床されるタスクが必要する処理量が加えられる。つまり、タスクの必要処理量もタイマーイベント実行に必要な処理量として計算される。

【0099】

ステップS12-10はタイマーイベントとその実行時間要求、許容範囲の登録である。ステップ12-11では、各タイマーイベントの時間要求と許容範囲を満たす範囲で、各タイマーイベントの必要処理量の単位時間辺りの合計が極力均等化するように実行時間を決定する。タイマーイベントの必要処理量にはタスクの起床の結果、タスクが必要とする分が加わっているので、ステップ12-11ではタイマーイベントとそれによって起床されるタスクの全ての処理量が均等化される。

【0100】

ステップS12-12は、CPU動作周波数の決定である。ステップS12-13は、登録されたタイマーイベントの実行である。タイマーイベントの実行は指定されている時間に起こるので、実際にはステップS12-13はステップS12-12の直後に実行されるわけではなく適切な時間の経過後に実行される。

【0101】

本実施の形態の実装としては、例えば、タイマーイベントを実行すべき時間ごとのキューで管理する。キューには、タイマーイベントを登録する際に、実行時間、許容範囲、必要処理量（タスク分を含む）を合わせて記録する（ステップS12-1からS12-5、S12-6からS12-10）。その上で、新しいタイマーイベントが登録される毎に、キューを走査し、各時間での必要処理量が均等化するように許容範囲内でキューを組替え、各時間でのCPU動作周波数を決定する（ステップS12-11、S12-12）。タイマーイベントを組替えられたキューに従った順番で実行し（ステップS12-13）、合わせて決定した各時間の動作周波数でCPUを駆動する。

【0102】

図13は本実施の形態での実際の処理の一例である。

【0103】

図13(a1)は、タイマーイベントが要求している実行時間にそのままタイマーイベントを配置した場合を表している。(a1)は、図12のステップS12-1からステップS12-5、ステップS12-6からステップS12-10まで実行された結果、タイマーイベント決定部に通知される情報である。四角のブロックはタイマーイベントを表している。(a1)では、A、B、Cの3つのタイマーイベントが要求されている。タイマーイベント名であるアルファベットの後の値はタイマーイベントの必要処理量であり、その後ろの値は実行時間要求と許容範囲である。タスクを起床するタイマーイベントの場合

は、タイマーイベント自身の処理量の後ろに、+でタスクの処理量を表記している。タイマーイベント A の場合、タイマーイベント自身の必要処理量が 50、起床されるタスクの処理量が 50、実行時間要求が 0、許容範囲が +2 となる。

【0104】

図 13 の例では、タイマーイベントとタスクは、実行時間要求の値の時間からその次の時間までの間隔での実効と処理の完了を要求しているとする。A の場合は、時刻 0 から 1 の間での実行と処理の完了を要求している。実施の形態 1 の例（図 3、図 4）と同様に、これを「時刻 0 での実行を要求している」という様に表記する。許容範囲の値も、同様に実行時間の変動の範囲を表している。

【0105】

図 13（b1）は、（a1）の実行時間でタイマーイベント A、B、C を実行し、各時間でそれぞれの処理を必要最小限の動作周波数で、アイドル期間をつくらないように CPU 動作周波数を設定した場合を表している。

【0106】

図 13（a2）は、ステップ S12-11 が実行されて決定されたタイマーイベントの実際の実行時間を表している。図 13（b2）は、（a2）のタイマーイベント実行時の CPU 動作周波数の設定である。（b1）と比較すると動作周波数の変動が少なくなっており、この期間での CPU の消費電力が削減される。また、各タイマーイベントはそれぞれの許容範囲を満たして実行されている。

【0107】

図 14 は、本実施の形態の実際の実処理の別の例である。図 14（a1）は、図 12 と同様に、タイマーイベント実行時間の要求である。タイマーイベント A は周期 3 での周期的実行、B は周期 2 での周期的実行を要求しており、図 14（a1）ではこれを実行時間要求が P3、P2 と表記している。タイマーイベント A、B はいずれもタスクを起動するが、タイマーイベント A で起動されるタスクは実行される時間が 1 遅れても許容できるので、タイマーイベント A も +1 の許容範囲を持つ。（a2）はプログラムの実行時間の決定後であり、この例では、タイマーイベント A の実行時間が要求している時間より変更されている。

【0108】

（b1）、（b2）は処理に必要な最小の動作周波数を用いる場合の動作周波数である。（b2）は完全な均等化ではないが（b1）より動作周波数の変動が少なくなり、CPU の消費電力は削減される。また、タイマーイベントとタスクは、それぞれの許容範囲を満たして実行されている。

【0109】

以上の様に、本実施の形態では、タスクとタイマーイベントの実行時間に関する要求と許容範囲内で満たした上で、CPU の消費電力を削減することが可能である。なお、他の実施の形態と同じく、全てのタイマーイベントが許容範囲を登録する必要は無い。処理量を平滑化されるのが全ての処理の一部であっても、その分の処理量は平滑化されるので、消費電力の削減効果を得ることが可能である。また、本実施の形態ではタスク及びタイマーイベントの両方の必要処理量を平滑化しているが、必ずしもその必要は無い。いずれかの必要な処理量が十分小さい場合は、影響が小さいとして無視しても良い。例えばタスクに比べてタイマーイベントの必要処理量が十分小さい場合は、タイマーイベントの必要処理量は実質 0 として、タスクの必要処理量のみを平滑化しても良い。

【0110】

また、本実施の形態においても実施の形態 2 と同様に、割り込み処理量検知部と、割り込み処理部と、割り込み発生時にタイマーイベント実行時間を再決定する割り込み時実行時間調整部を設けることも可能である。この場合は、割り込み発生時に、割り込み処理の分を加えた上で処理量が均等化するようにタイマーイベントの実行時間を再決定する。これにより、割り込みの発生時にも対応することが出来る。

【0111】

なお、本発明の実施の形態ではOS内に、タスク実行時間登録部401、タスク実行時間要求解釈部402、タスク処理量検知部403、タイマーイベント処理量検知部404、タイマーイベント登録部405、タイマーイベント決定部406、CPU動作周波数設定部407、タイマーイベント実行部408、タスク実行部409を設けたが、OS外部に設けることも可能である。

【産業上の利用可能性】

【0112】

本発明にかかるCPUクロック制御装置及び方法は、情報処理装置を用いるものであれば幅広い分野において有効である。例えば、大型計算機やパーソナルコンピュータのような形態のみならず、各種の家電機器、携帯電話のような通信機器、産業機器、乗用機器などでも利用可能である。

【図面の簡単な説明】

【0113】

【図1】 本発明の実施の形態1におけるCPUクロック制御装置を構築する情報処理装置の概略構成図

【図2】 本発明の実施の形態1におけるCPU動作周波数制御時の動作を説明するフローチャート

【図3】 本発明の実施の形態1における処理の一例を説明する図

【図4】 本発明の実施の形態1における処理の一例を説明する図

【図5】 本発明の実施の形態2におけるCPUクロック制御装置を構築する情報処理装置の概略構成図

【図6】 本発明の実施の形態2におけるCPU動作周波数制御時の動作を説明するフローチャート

【図7】 本発明の実施の形態2における処理の一例を説明する図

【図8】 本発明の実施の形態3におけるCPUクロック制御装置を構築する情報処理装置の概略構成図

【図9】 本発明の実施の形態3におけるCPU動作周波数制御時の動作を説明するフローチャート

【図10】 本発明の実施の形態3における処理の一例を説明する図

【図11】 本発明の実施の形態4におけるCPUクロック制御装置を構築する情報処理装置の概略構成図

【図12】 本発明の実施の形態4におけるCPU動作周波数制御時の動作を説明するフローチャート

【図13】 本発明の実施の形態4における処理の一例を説明する図

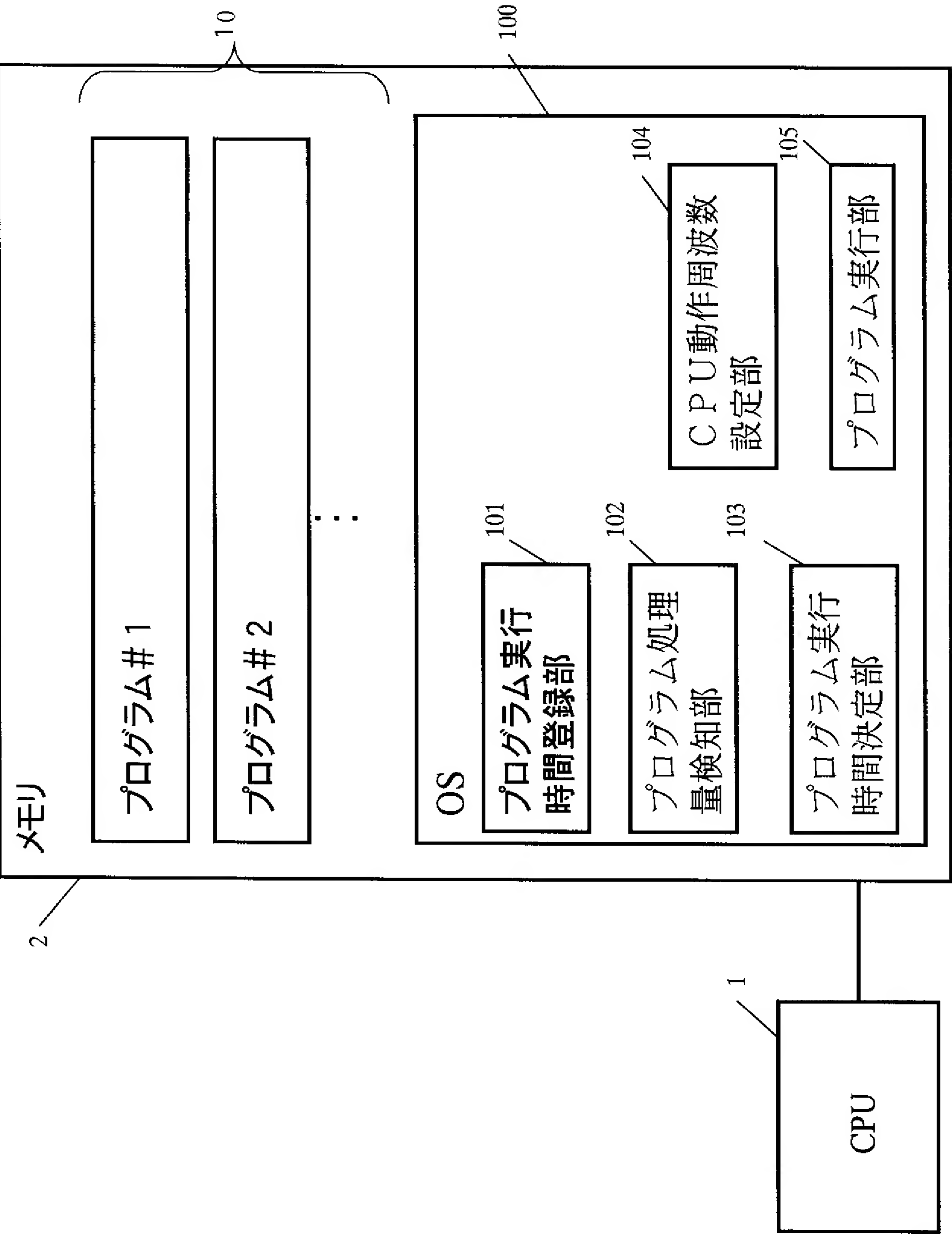
【図14】 本発明の実施の形態4における処理の一例を説明する図

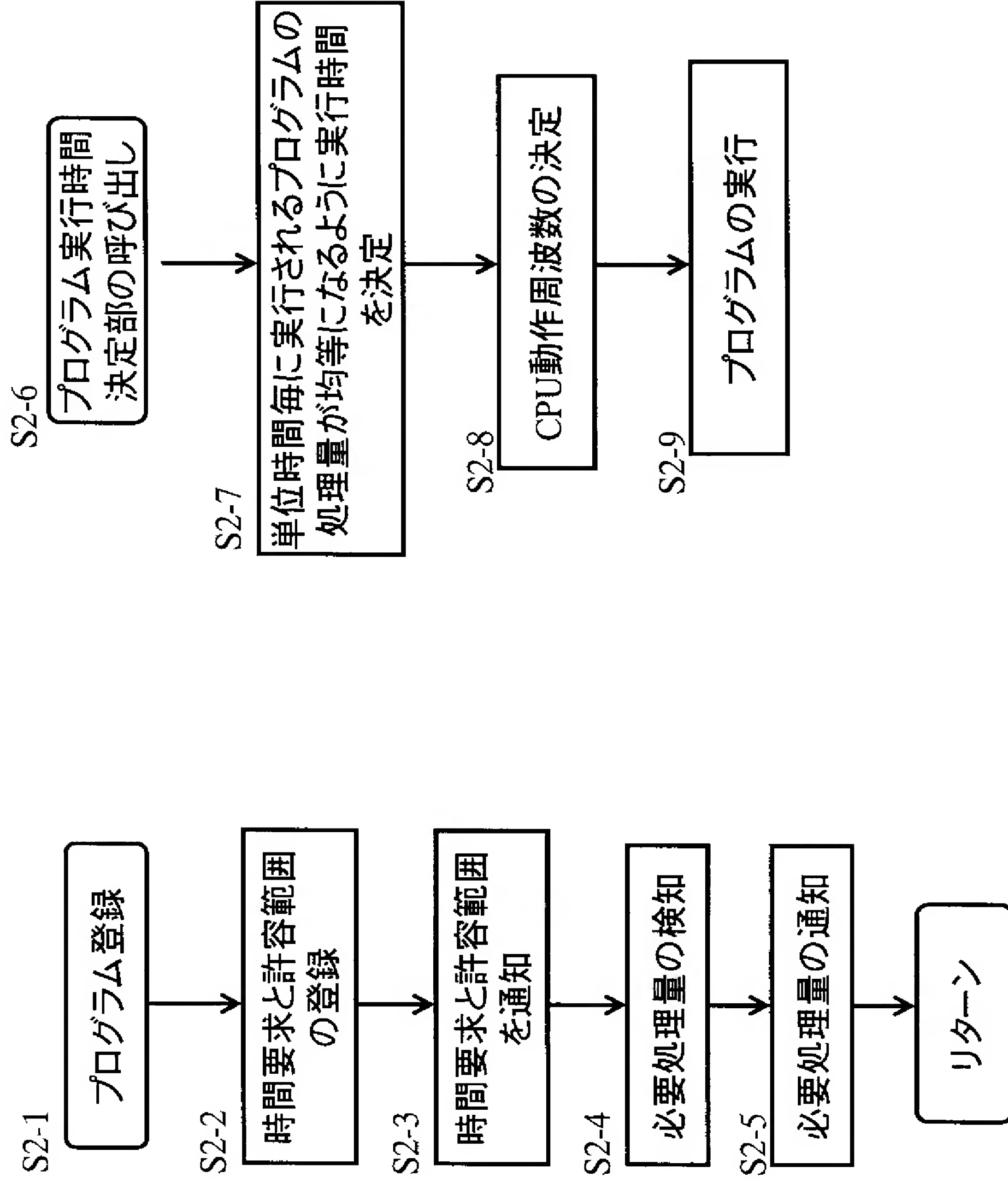
【符号の説明】

【0114】

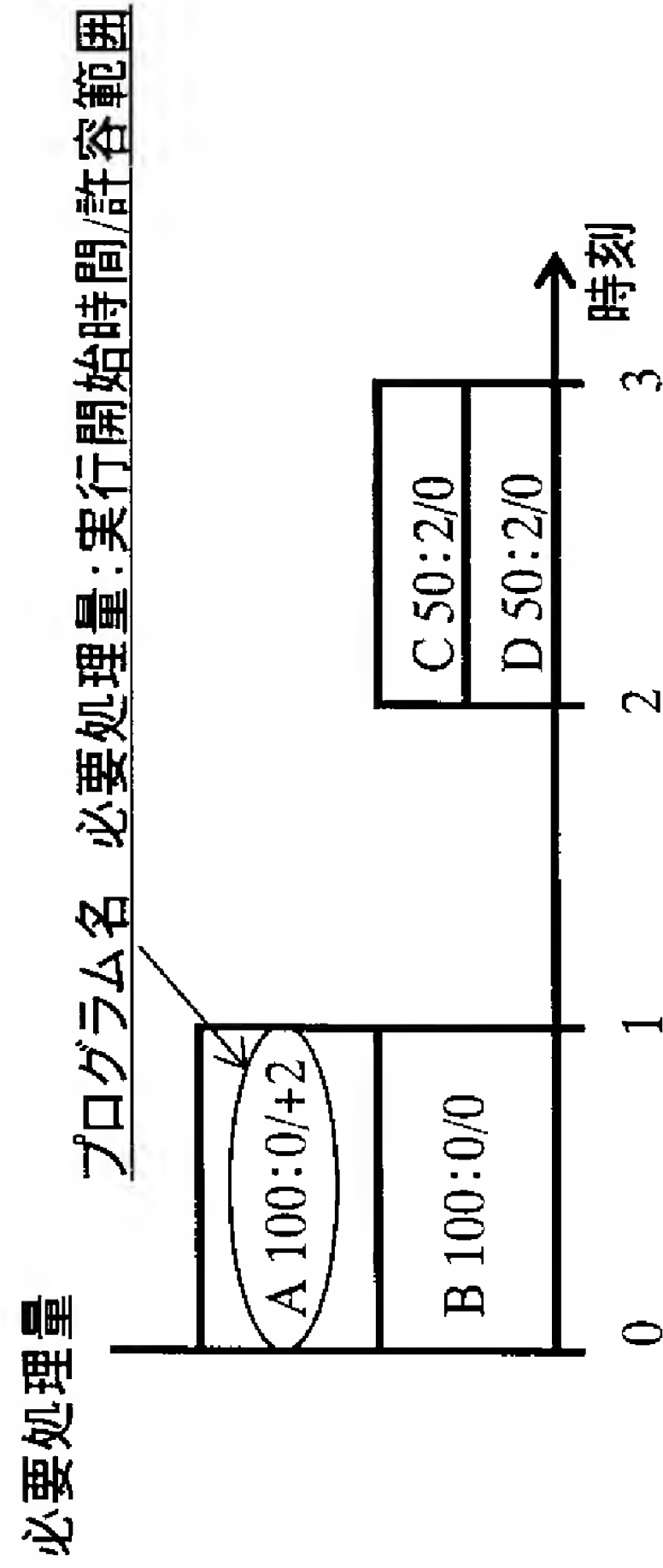
- 1 中央演算装置（CPU）
- 2 メモリ装置
- 10, 20 プログラム
- 30, 40 タスク
- 41 タイマーイベント
- 100, 200, 300, 400 オペレーティングシステム（OS）
- 101, 201 プログラム実行時間登録部
- 102, 202 プログラム処理量検知部
- 103, 203 プログラム実行時間決定部
- 104, 207, 304, 407 CPU動作周波数設定部
- 105, 208 プログラム実行部
- 204 割り込み処理量検知部
- 205 割り込み処理部

2 0 6	割り込み時実行時間調整部
3 0 1 , 4 0 1	タスク実行時間登録部
3 0 2 , 4 0 3	タスク処理量検知部
3 0 3	タスクスケジューリング部
3 0 5 , 4 0 9	タスク実行部
4 0 2	タスク実行時間要求解釈部
4 0 4	タイマーイベント処理量検知部
4 0 5	タイマーイベント登録部
4 0 6	タイマーイベント決定部
4 0 8	タイマーイベント実行部



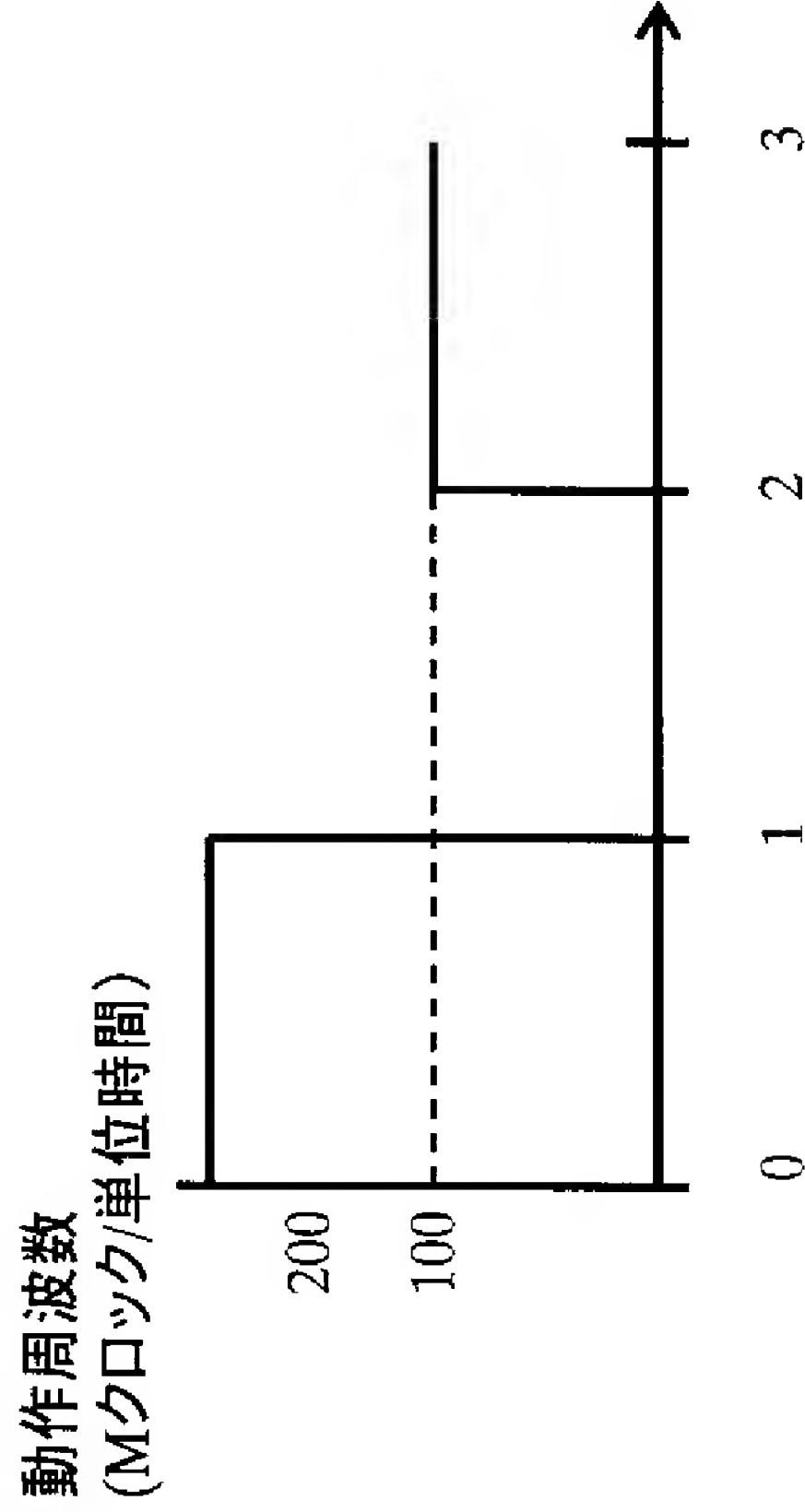


(a1) [プログラム登録時の実行時間要求]

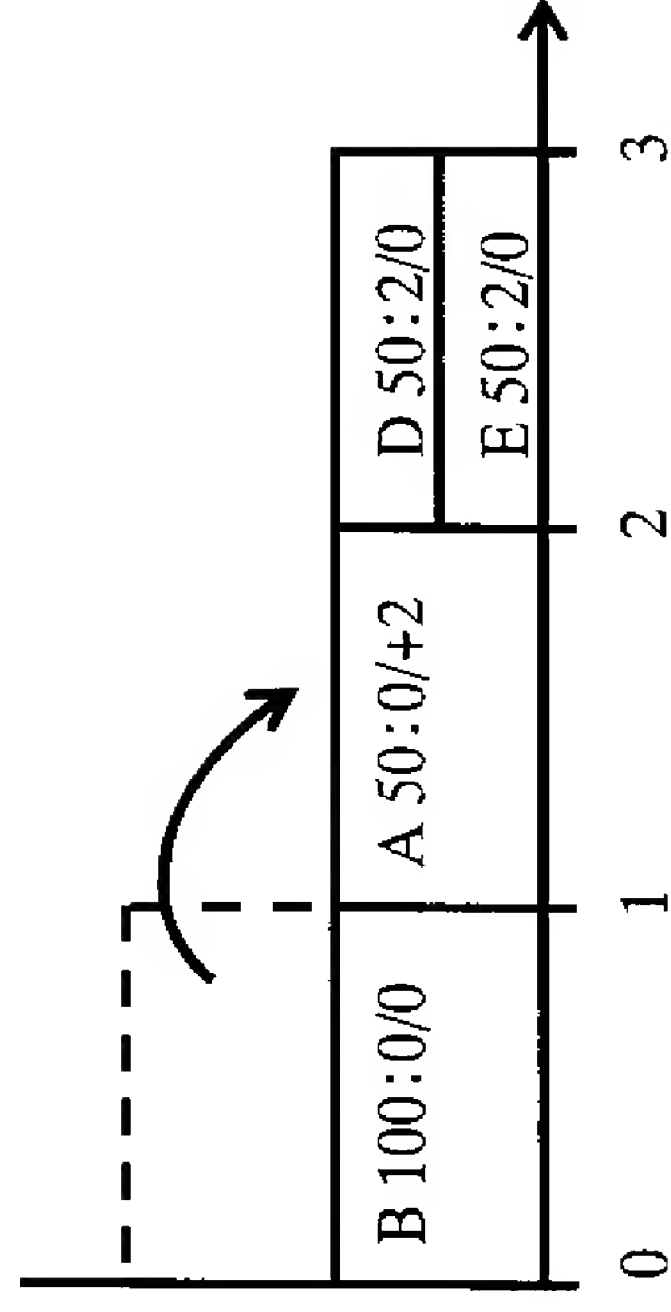


プログラムの決定

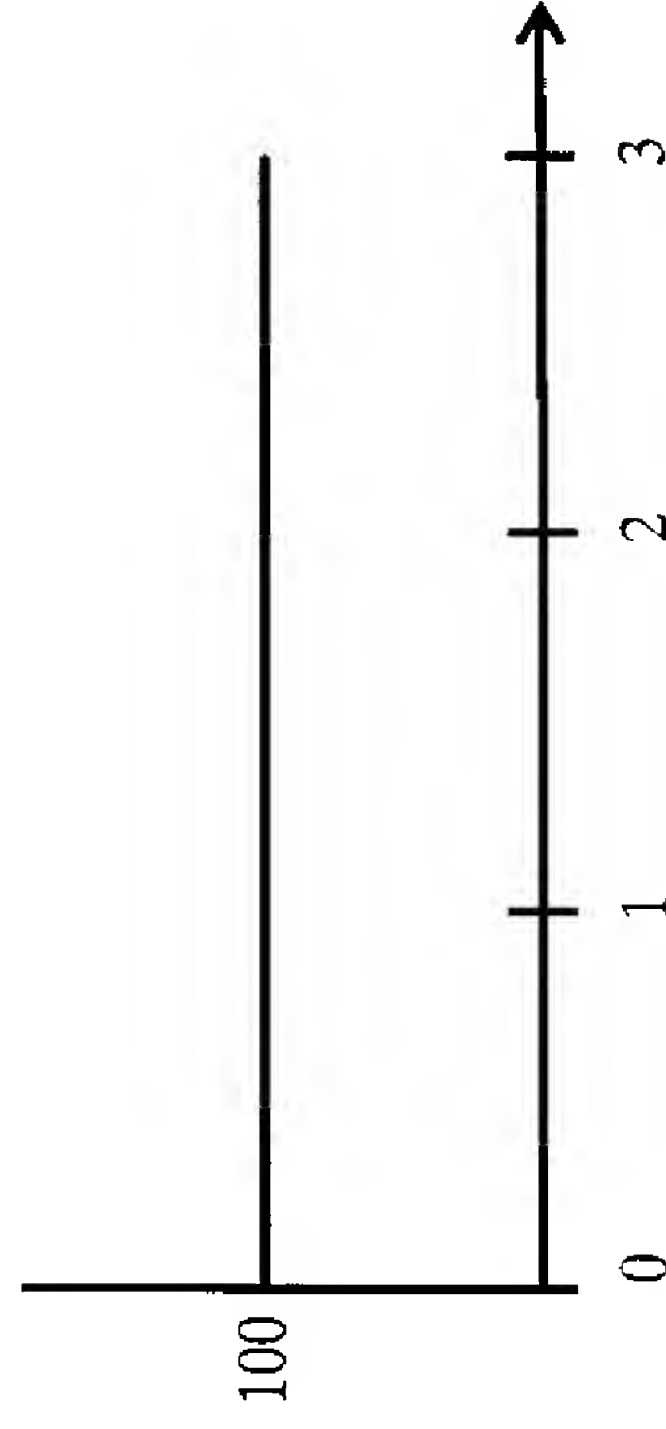
(b1) [CPU動作周波数(要求のまま実行した場合)]



(a2) [決定されたプログラム実行時間]

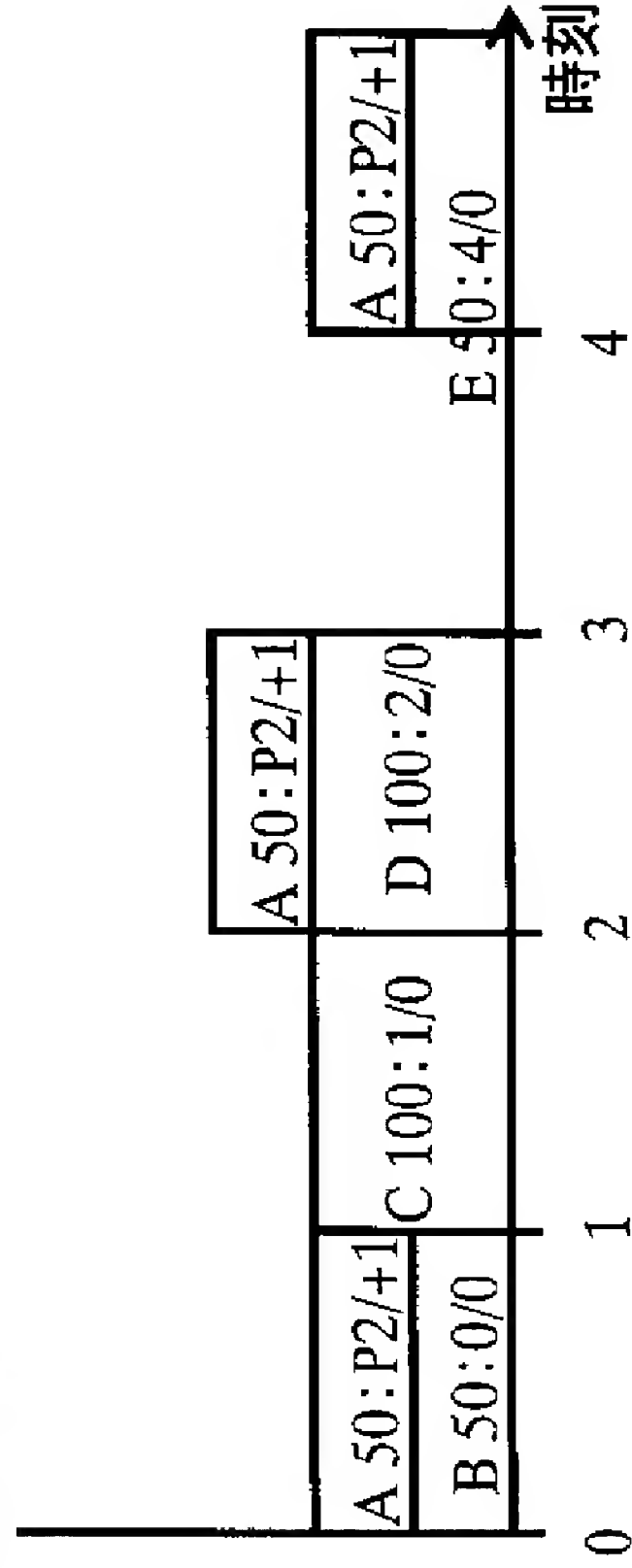


(b2) [CPU動作周波数(実行時間決定後)]

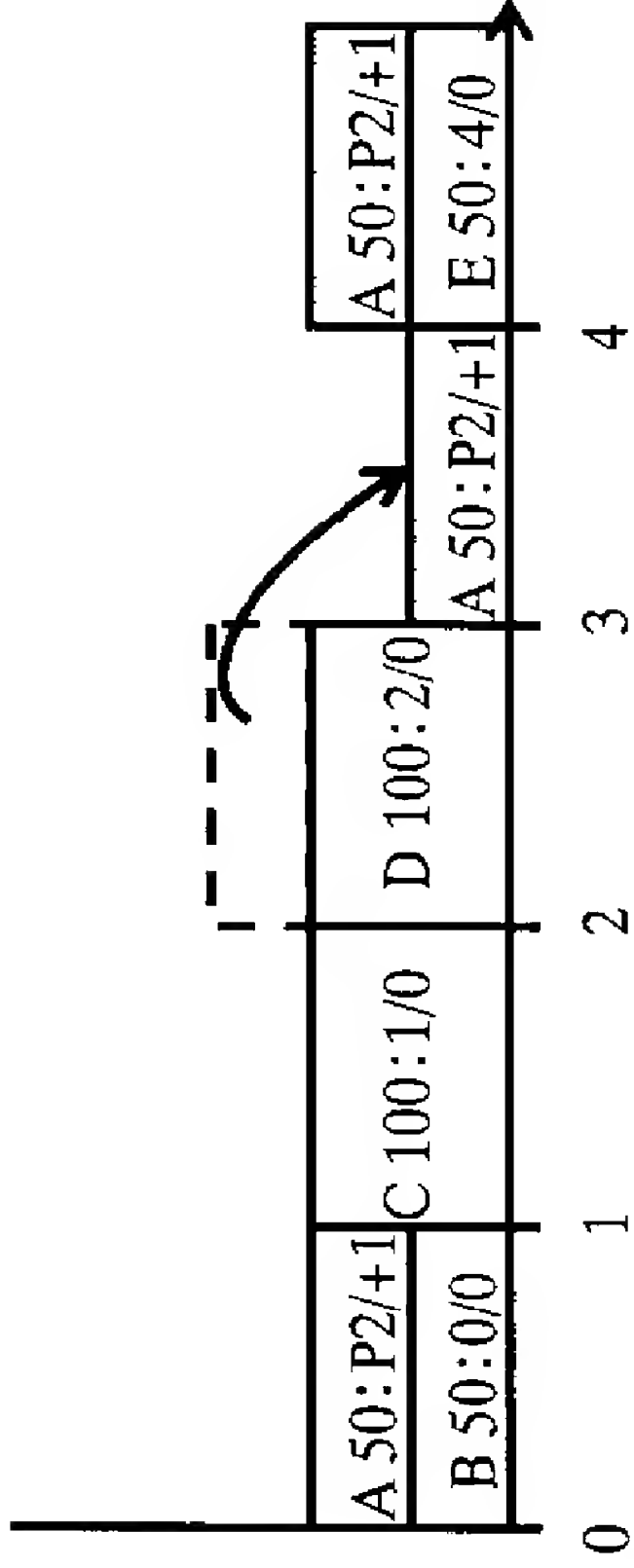


(a1) [プログラム登録時の実行時間要求]

必要処理量

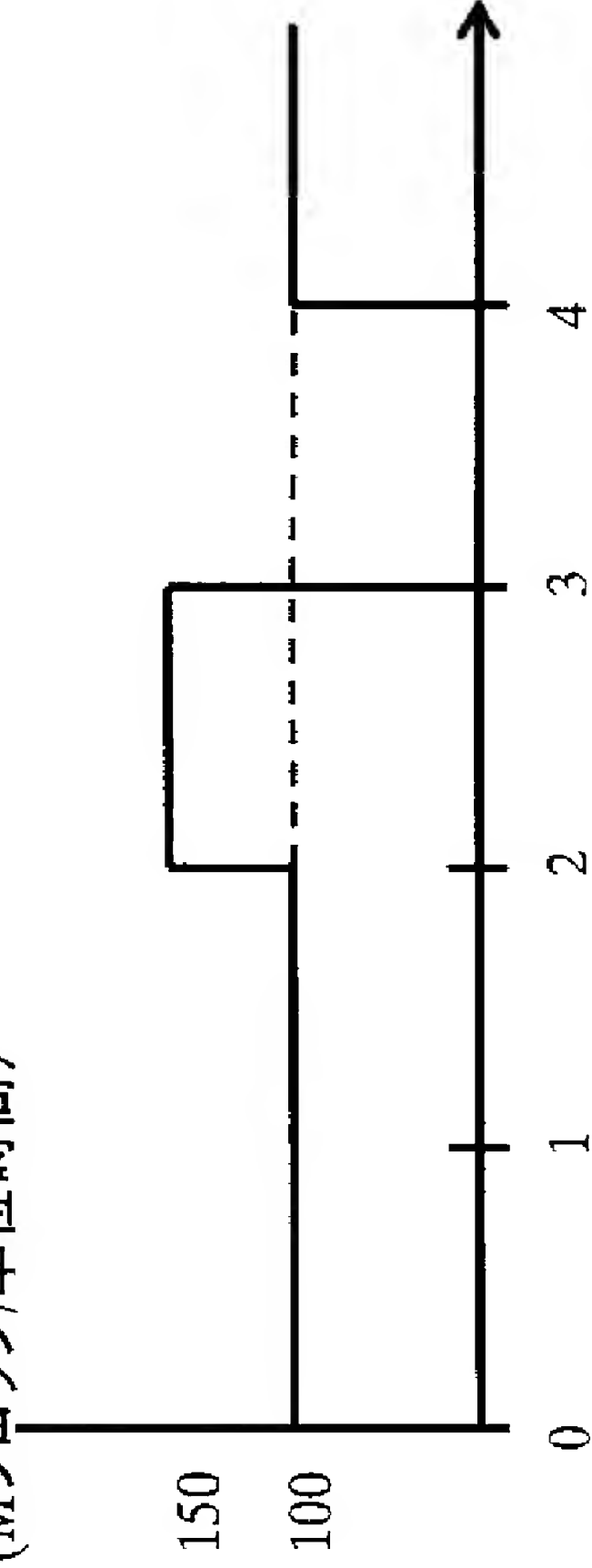


(a2) [決定されたプログラム実行時間]



(b1) [CPU動作周波数(要求のまま実行した場合)]

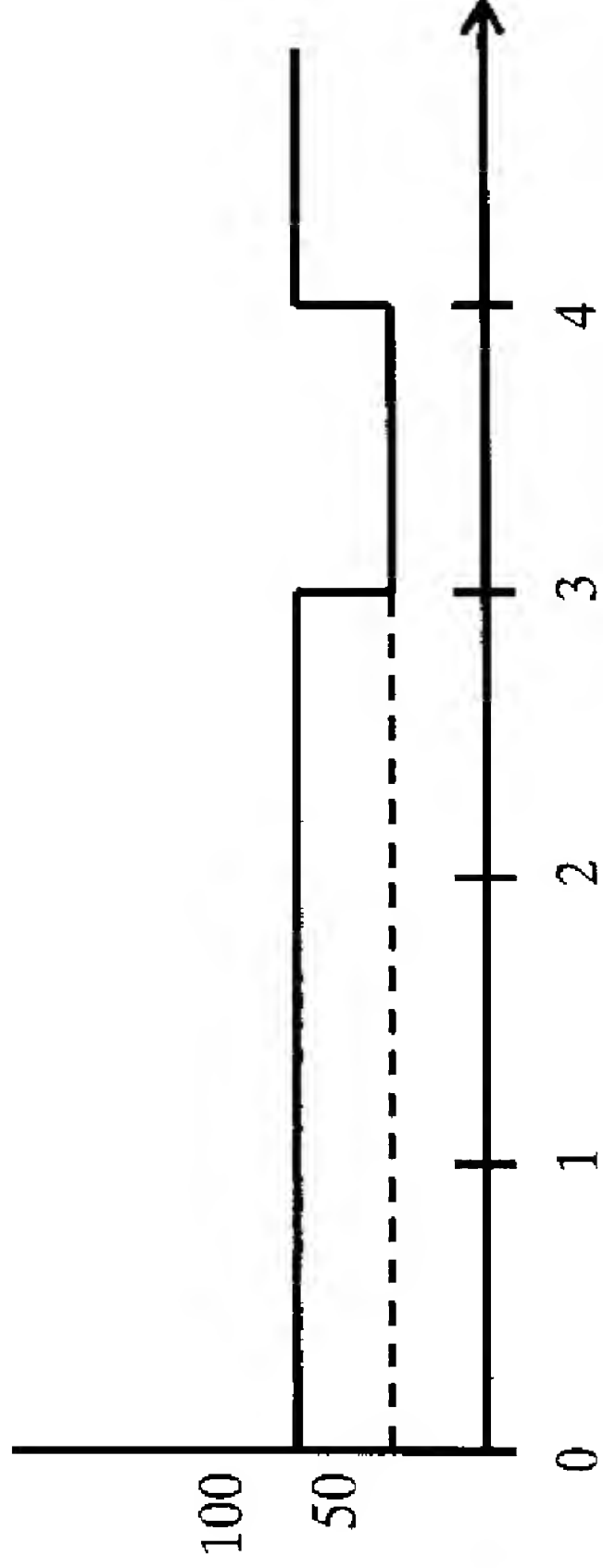
動作周波数
(MHz/単位時間)



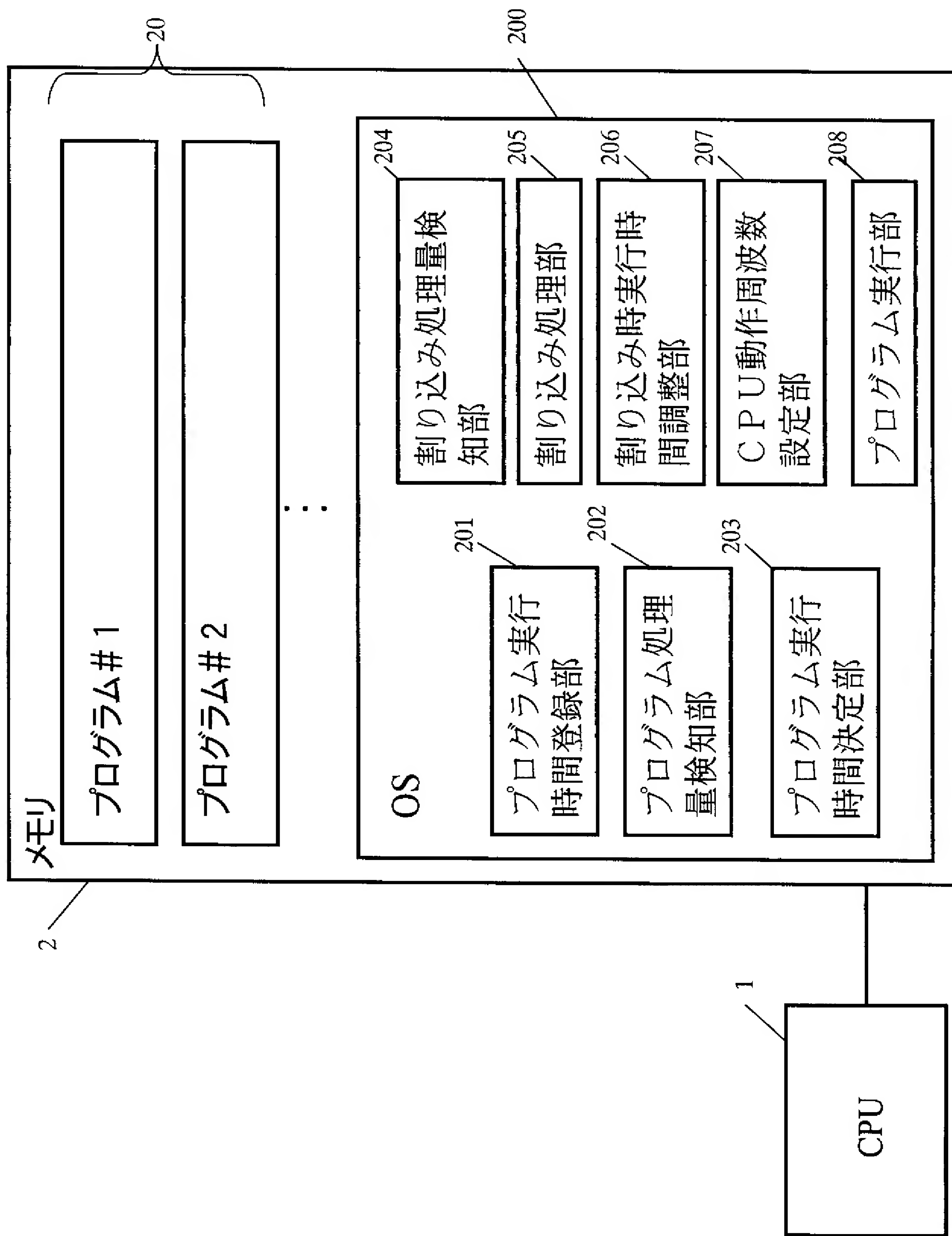
プログラムの実行時間

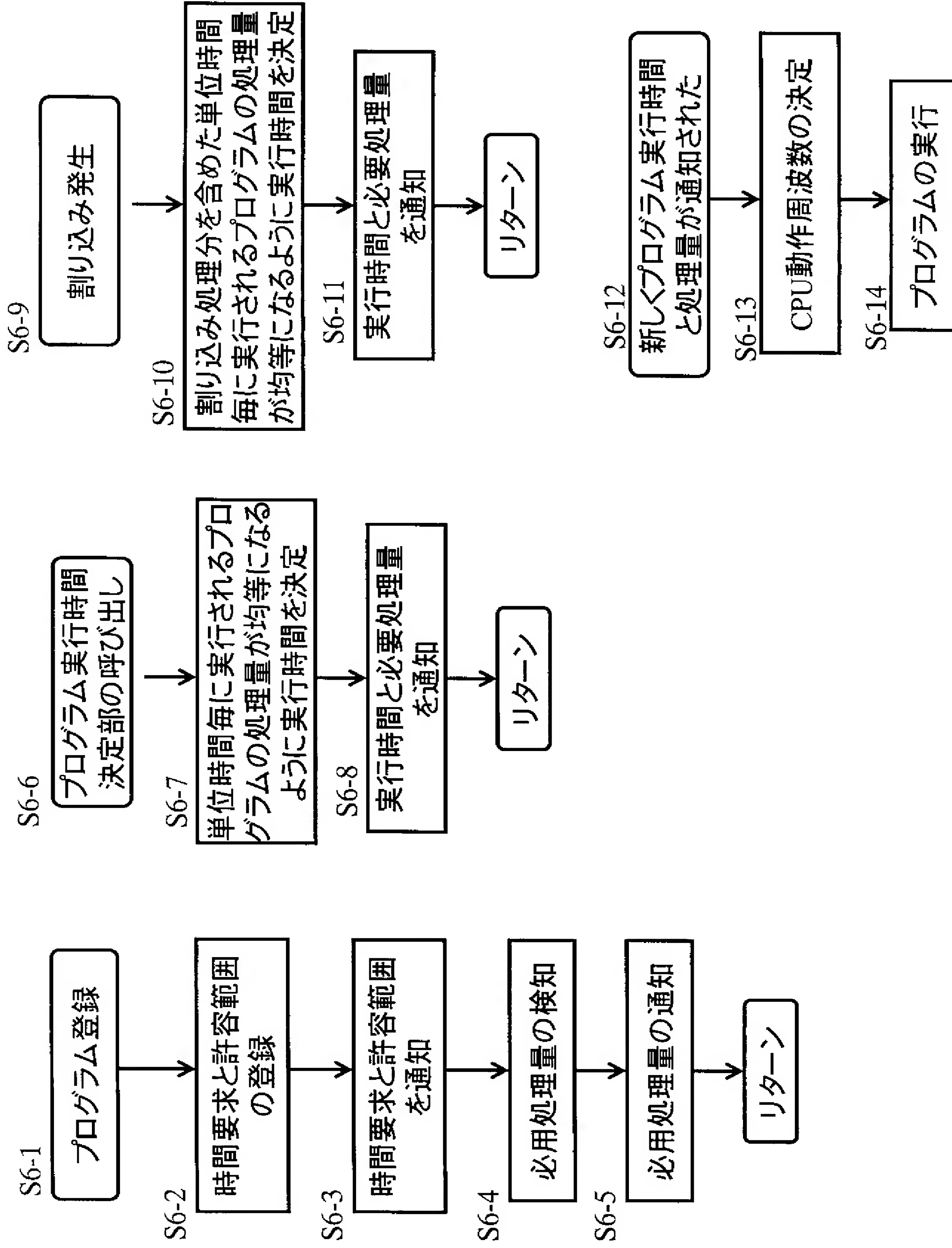
の決定

(b2) [CPU動作周波数(実行時間決定後)]

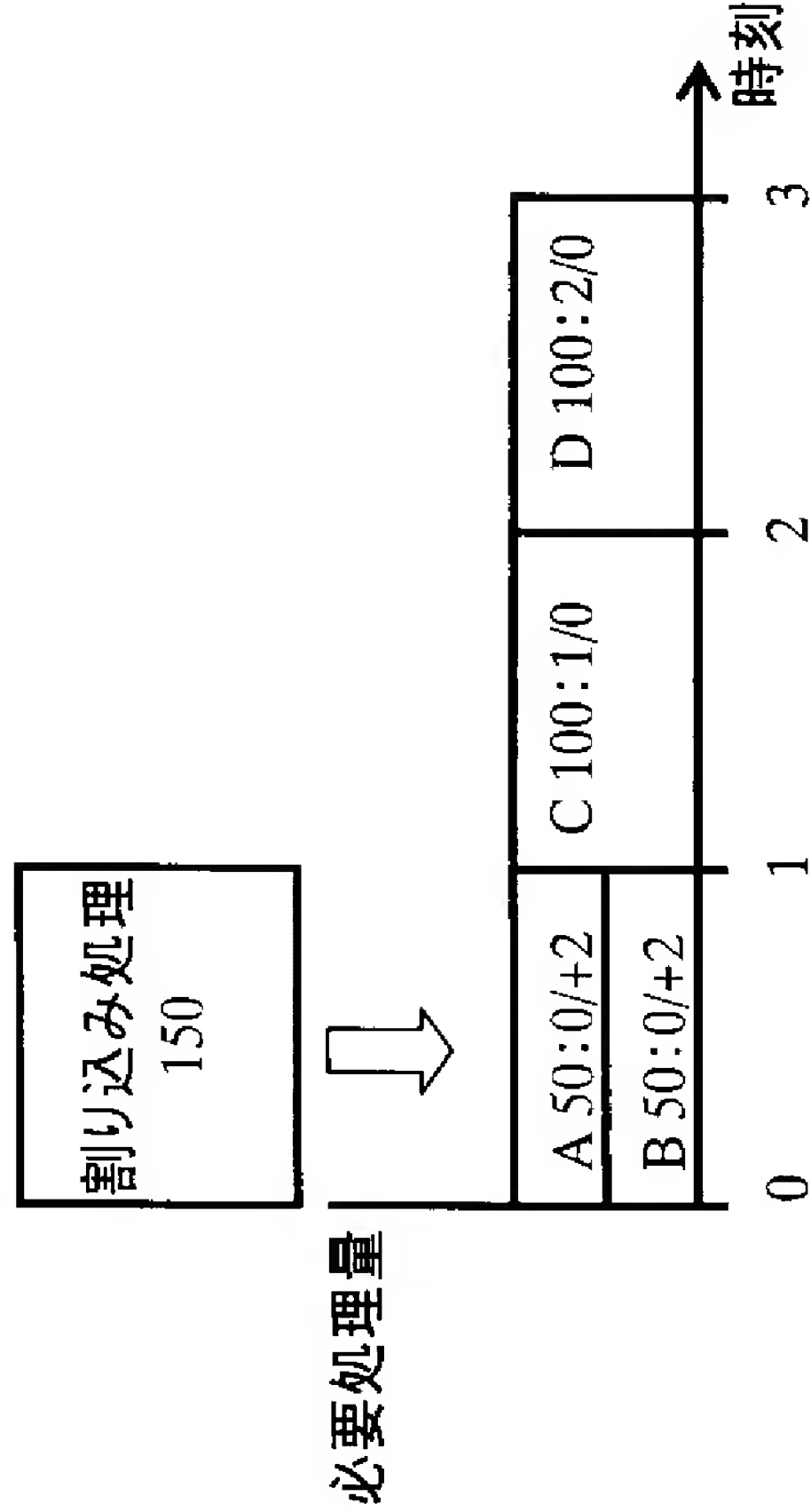


【図 5】

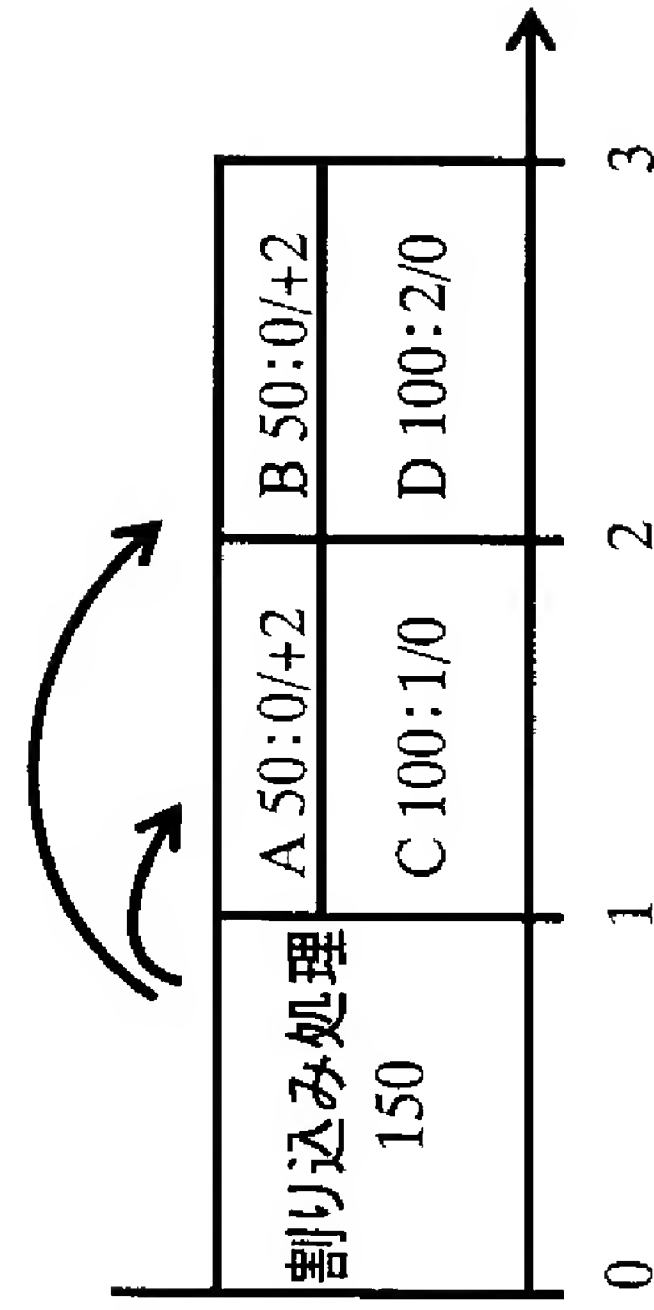




(a1) [割り込み発生]



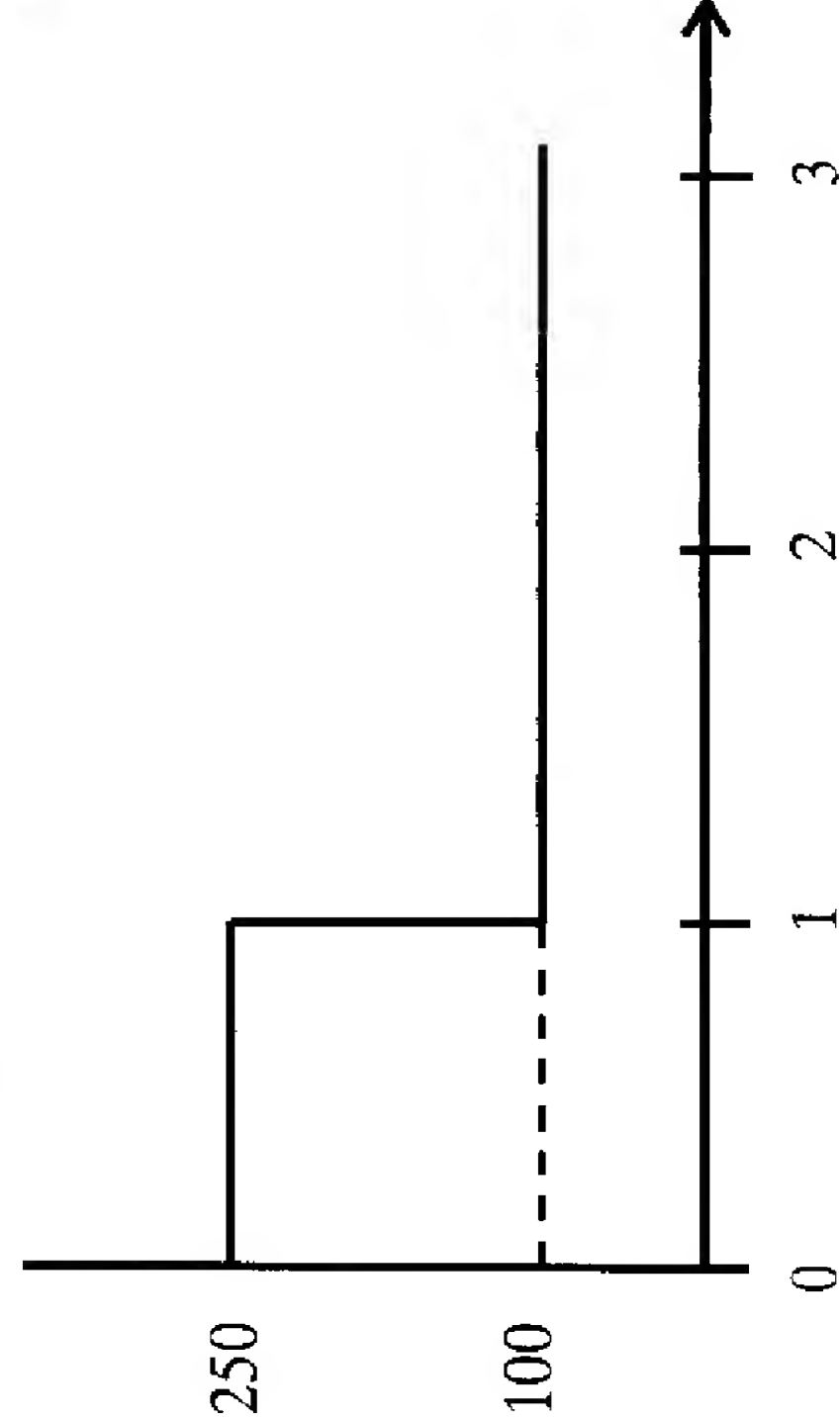
(a2) [再決定後のプログラム実行時間]



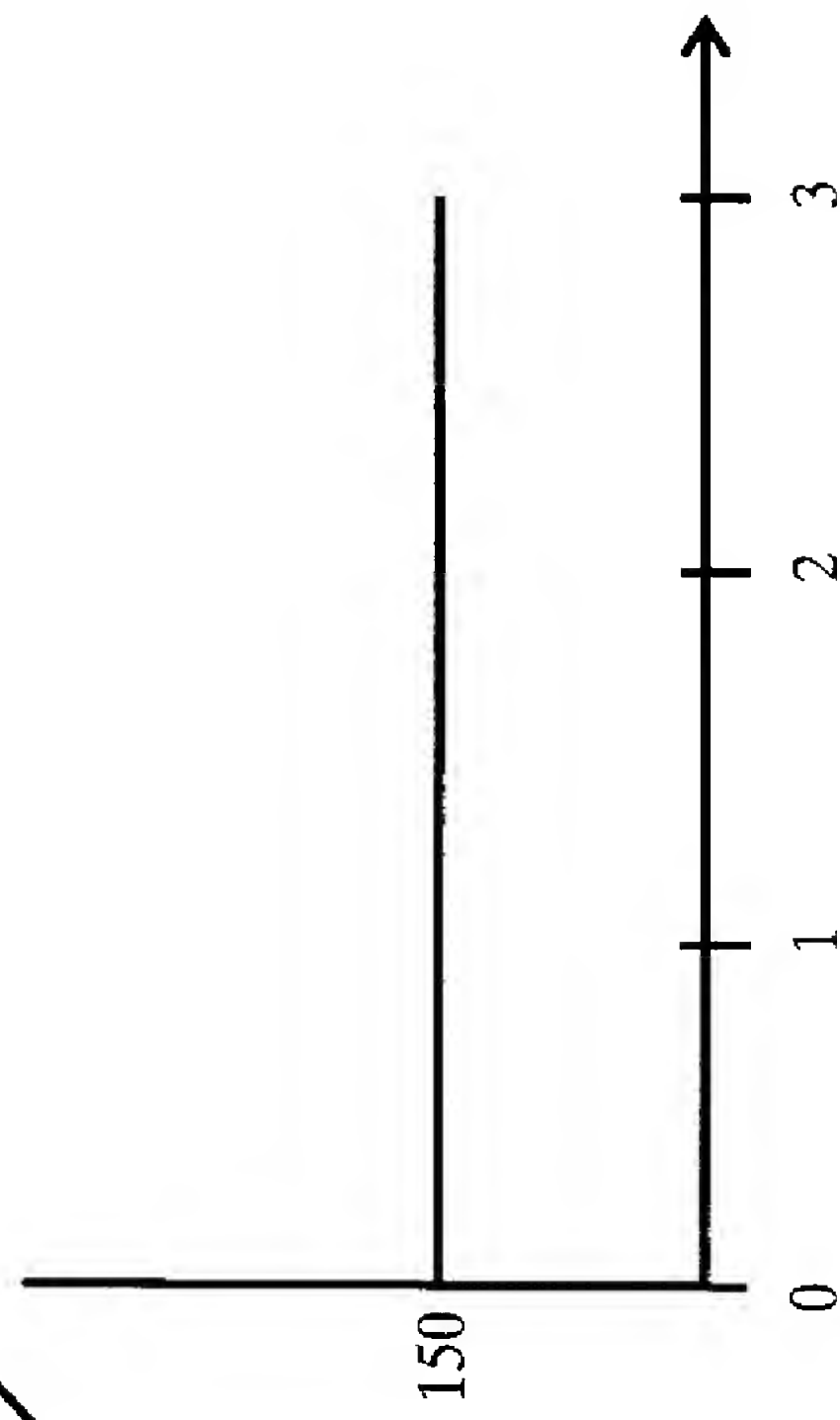
プログラム実行時間
の再決定

(b1) [CPU動作周波数(再決定前)]

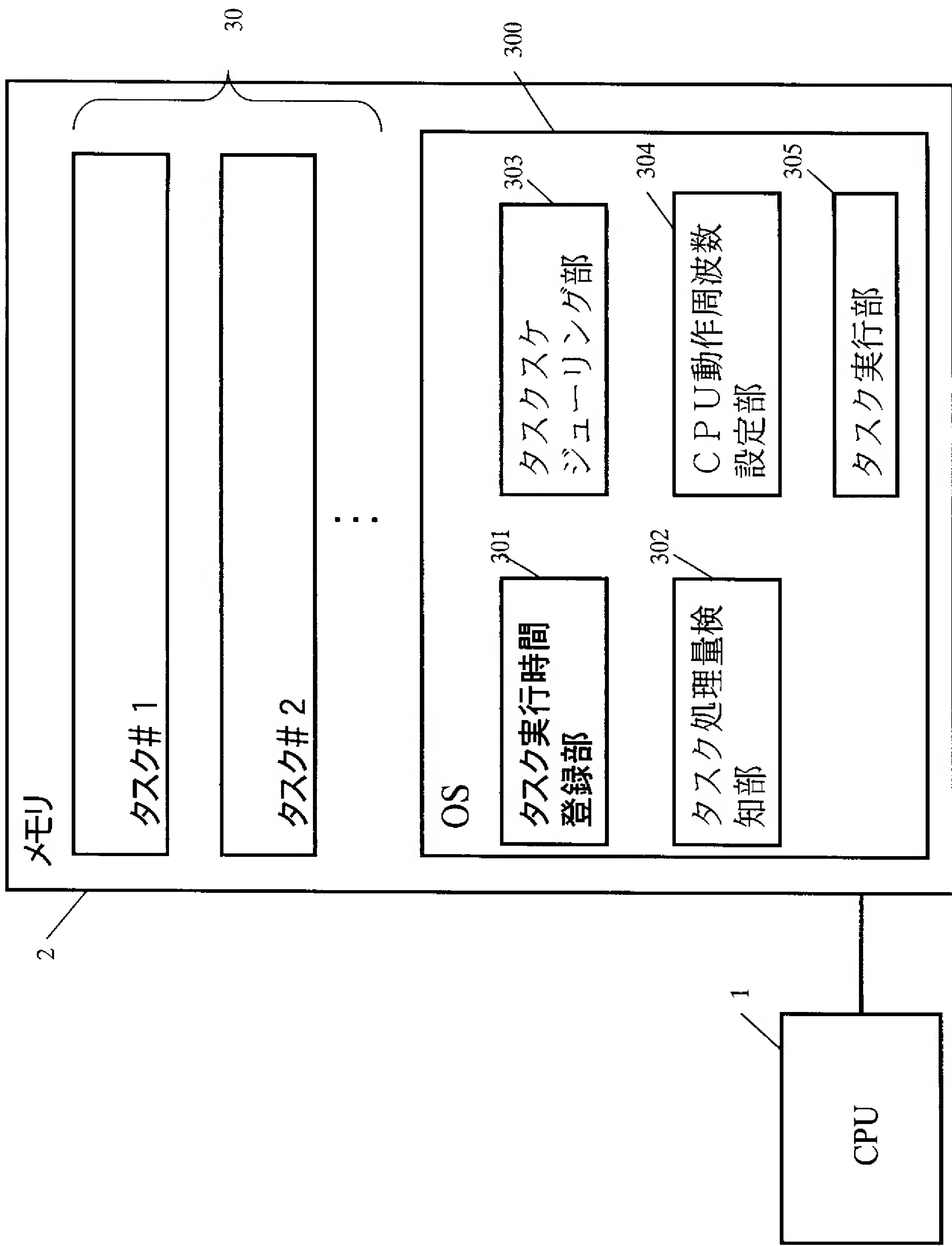
動作周波数
(Mクロック/単位時間)

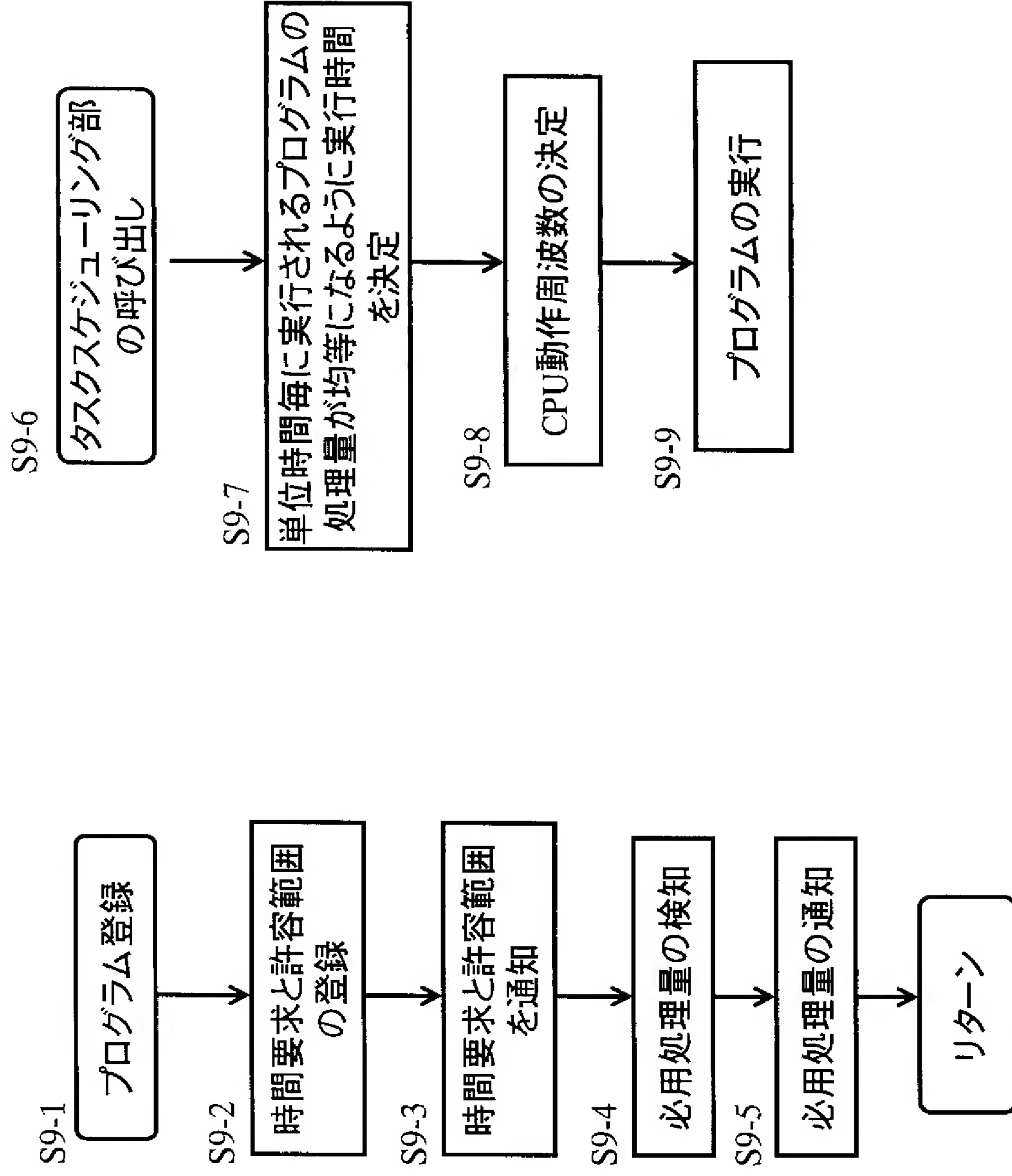


(b2) [CPU動作周波数(再決定後)]

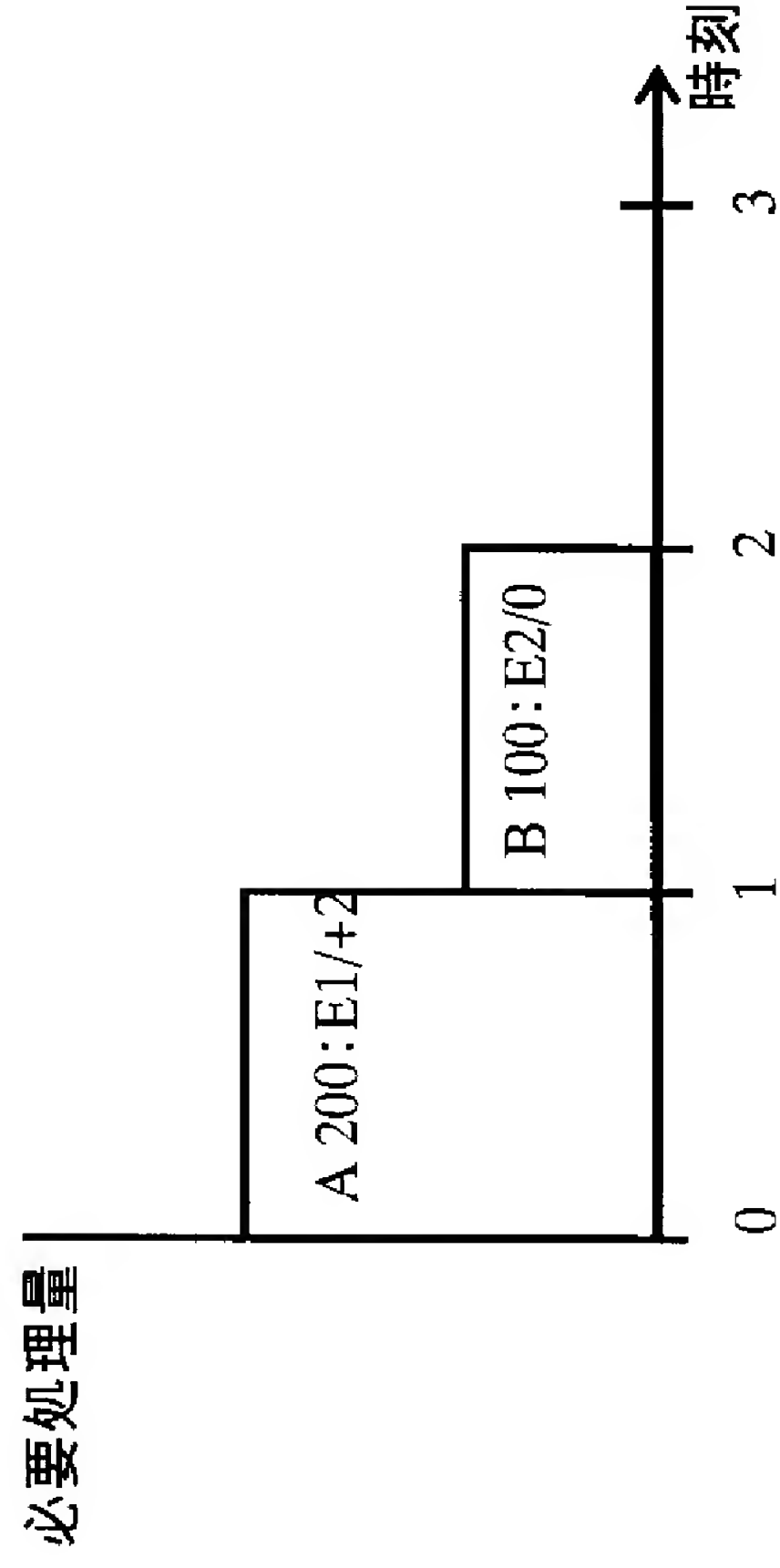


【図 8】

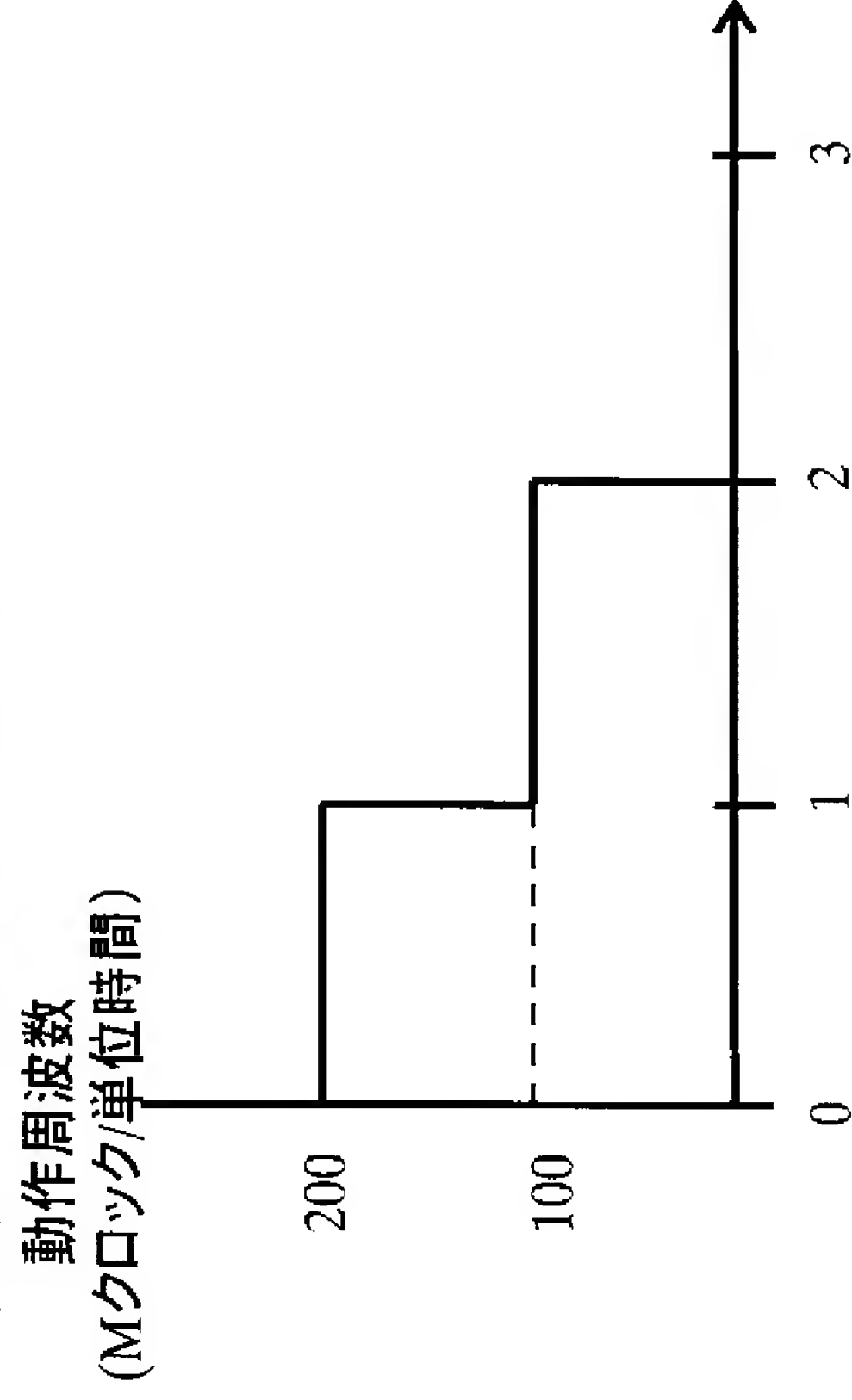




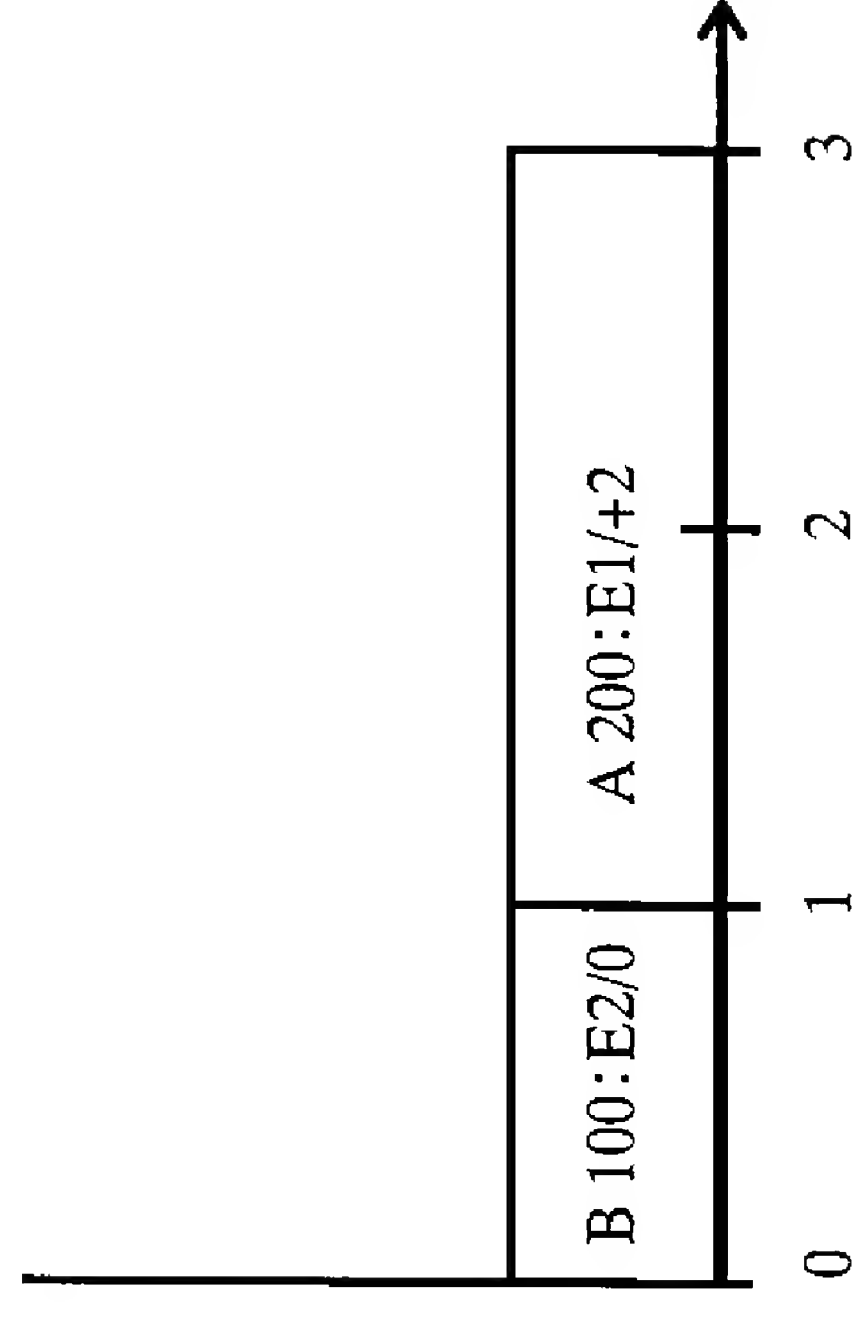
(a1) [通常のデッドラインスケジュール]



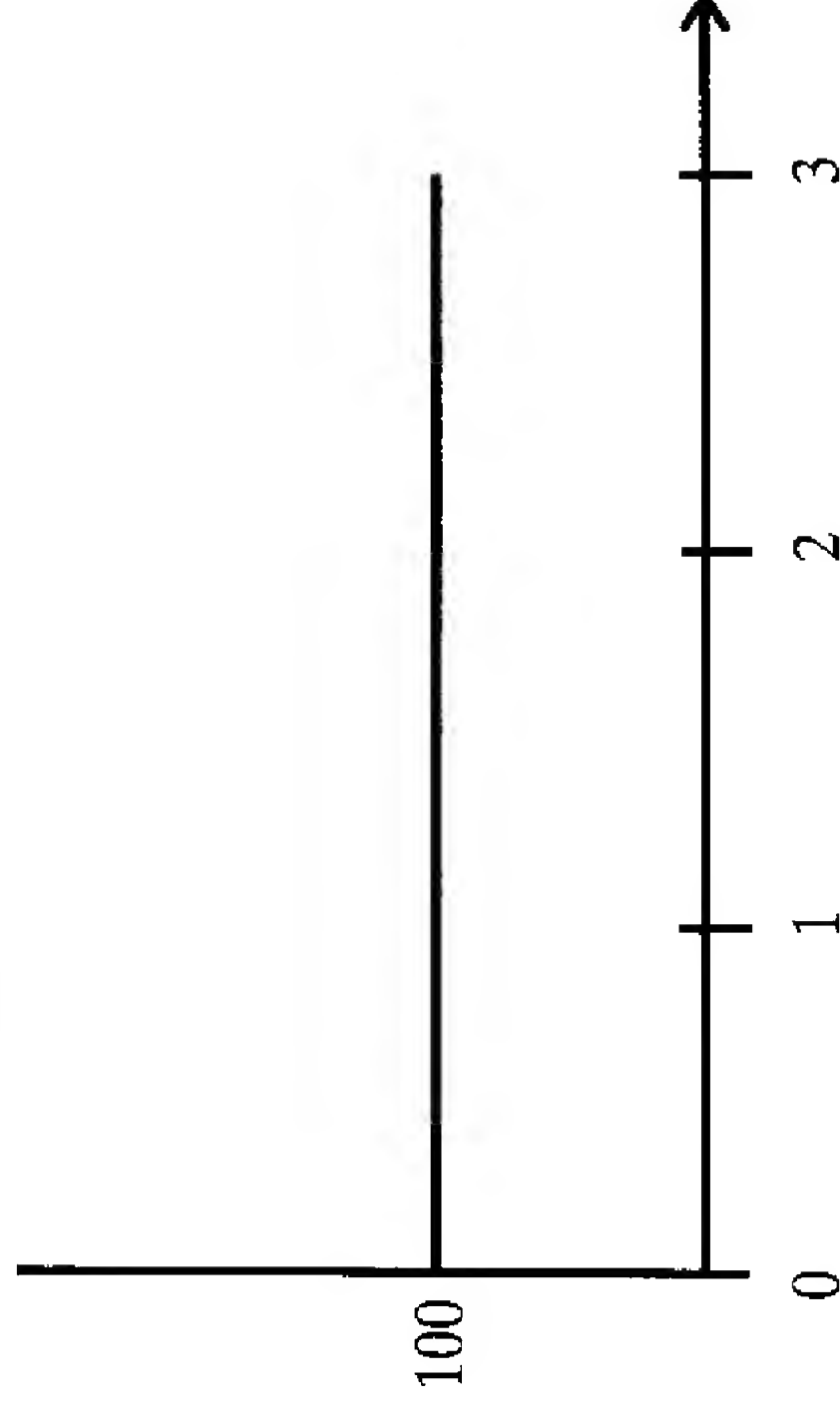
(b1) [CPU動作周波数(通常)]



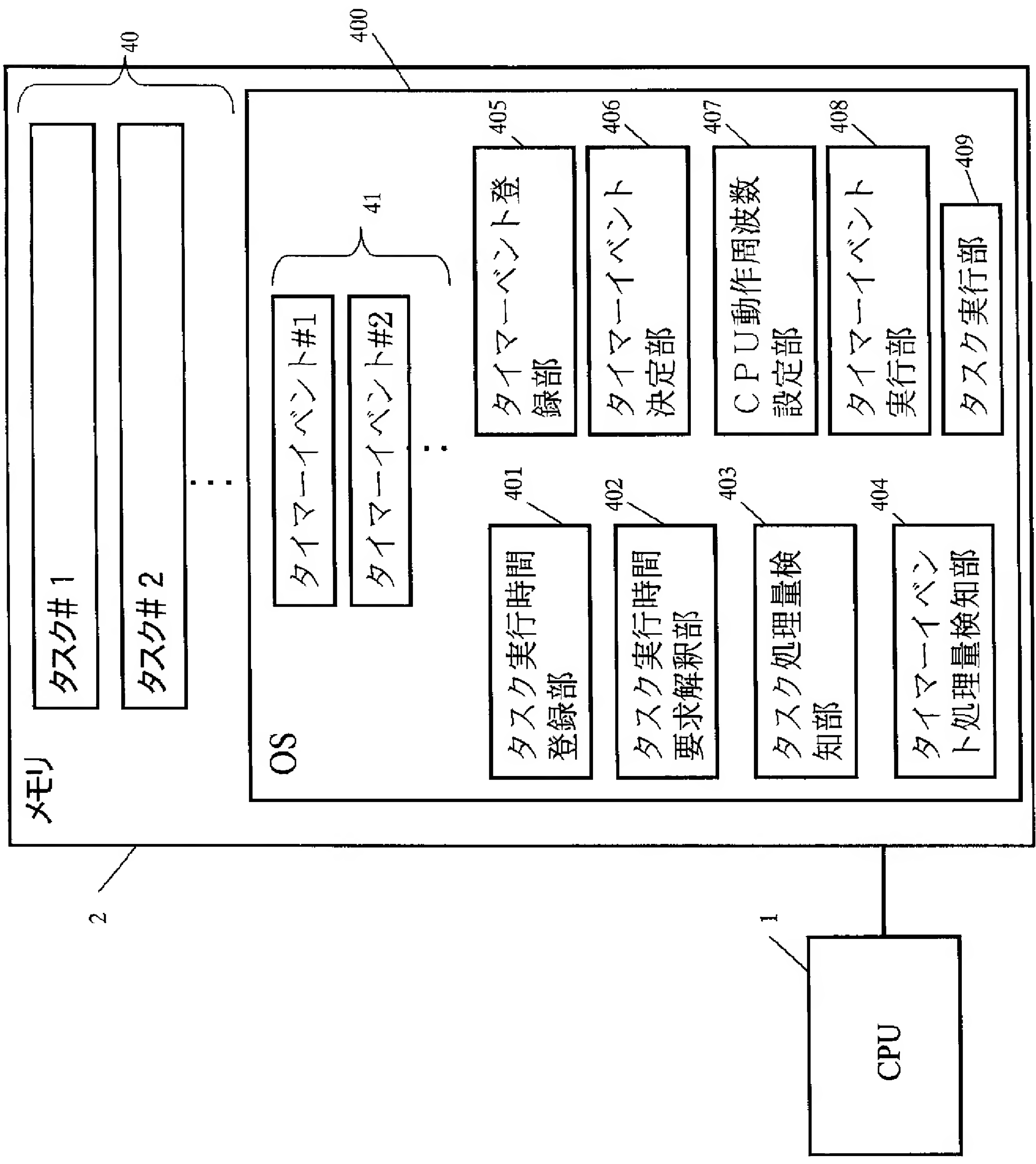
(a2) [許容範囲を用いたスケジューリング]

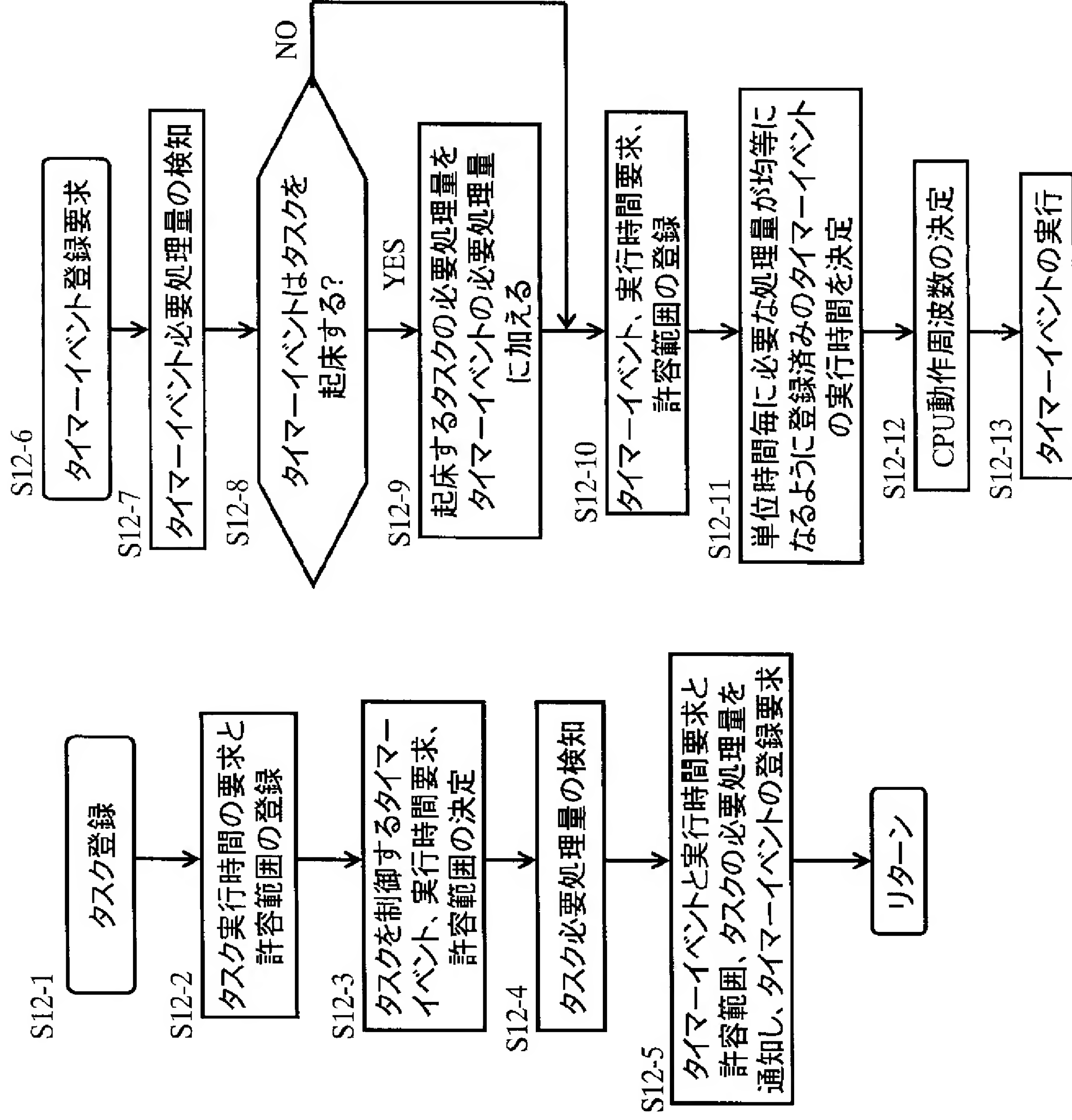


(b2) [CPU動作周波数(許容範囲を用いたスケジューリング)]



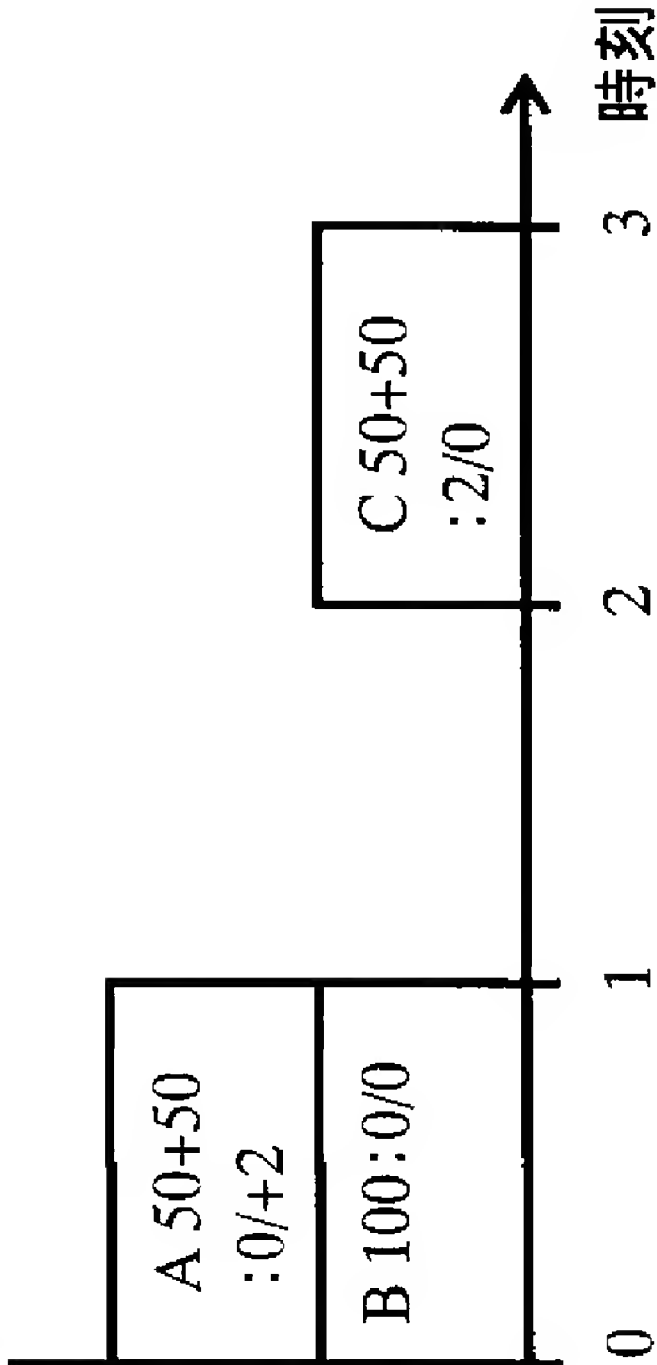
【図 1 1】



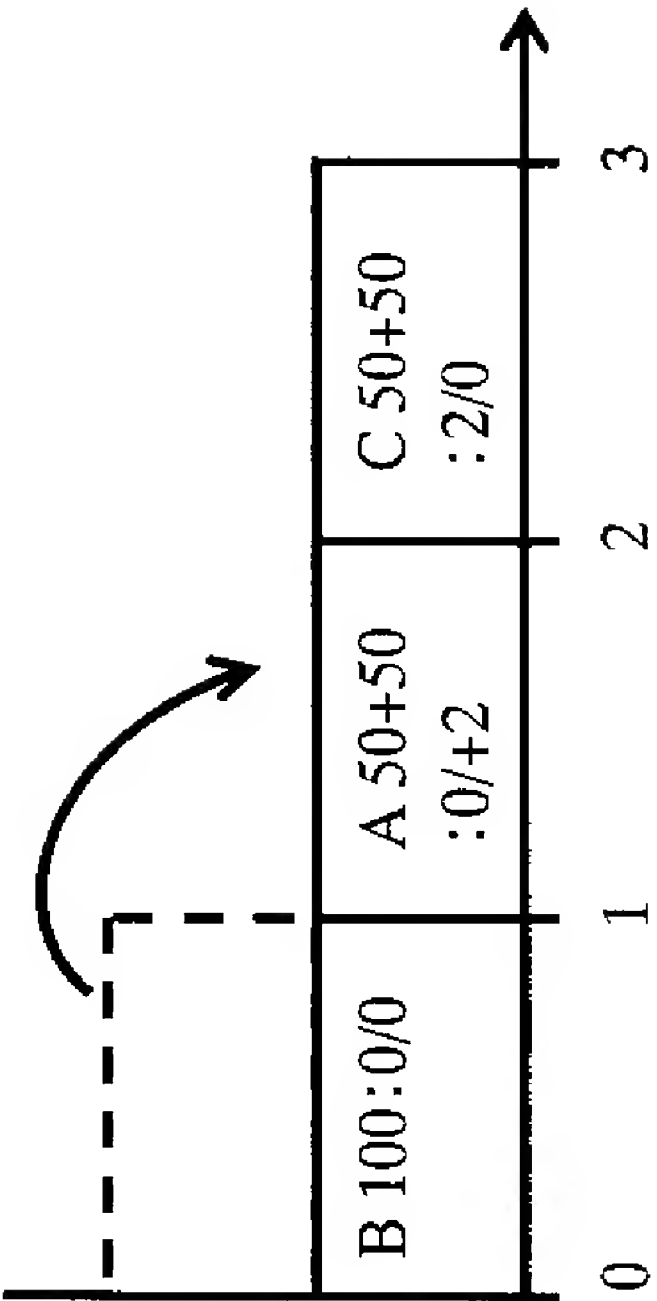


(a1) [タイマイベント登録時の実行時間要求]

必要処理量



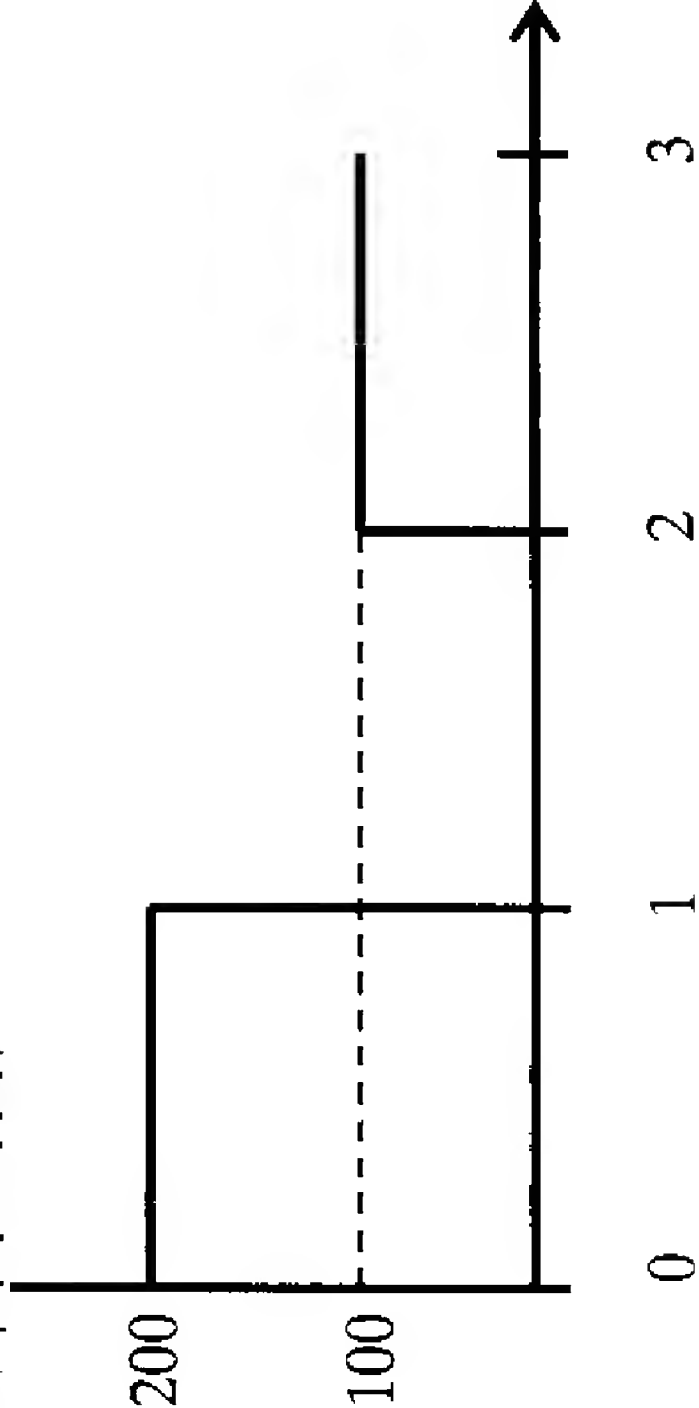
(a2) [決定されたタイマイベント実行時間]



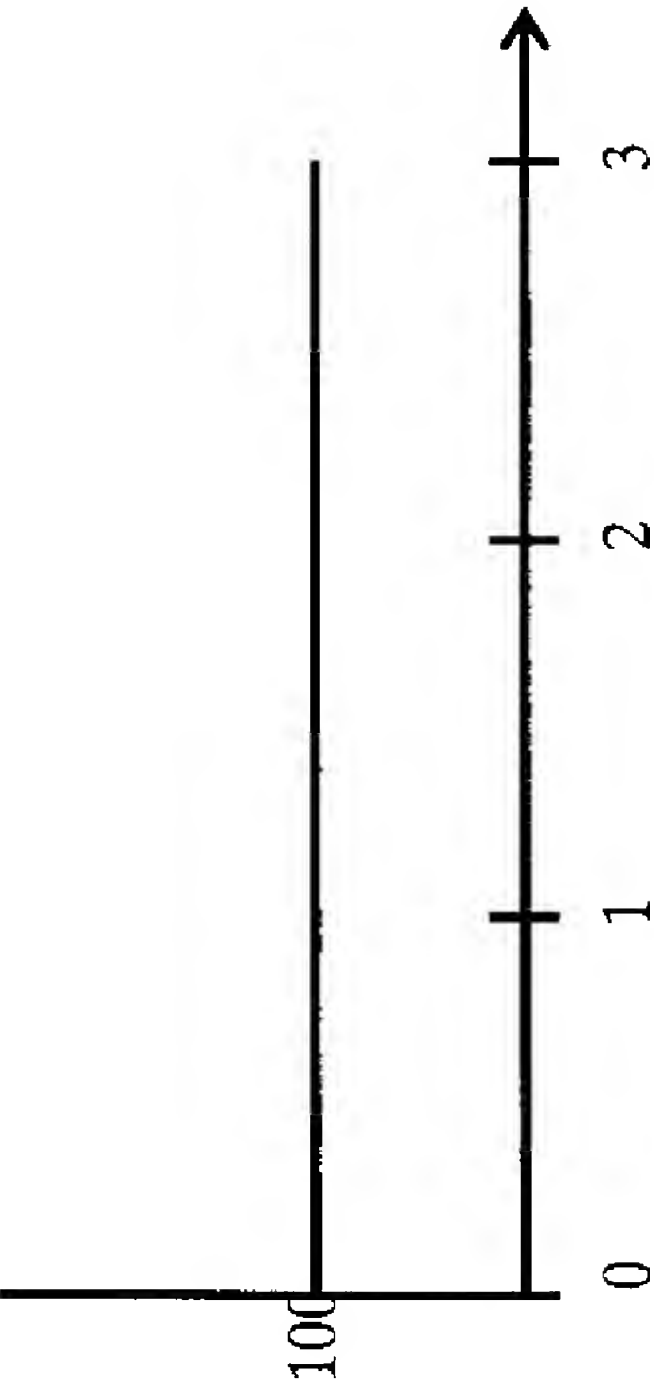
タイマイベント
実行時間の決定

(b1) [CPU動作周波数(要求のまま実行した場合)]

動作周波数
(Mクロック/単位時間)



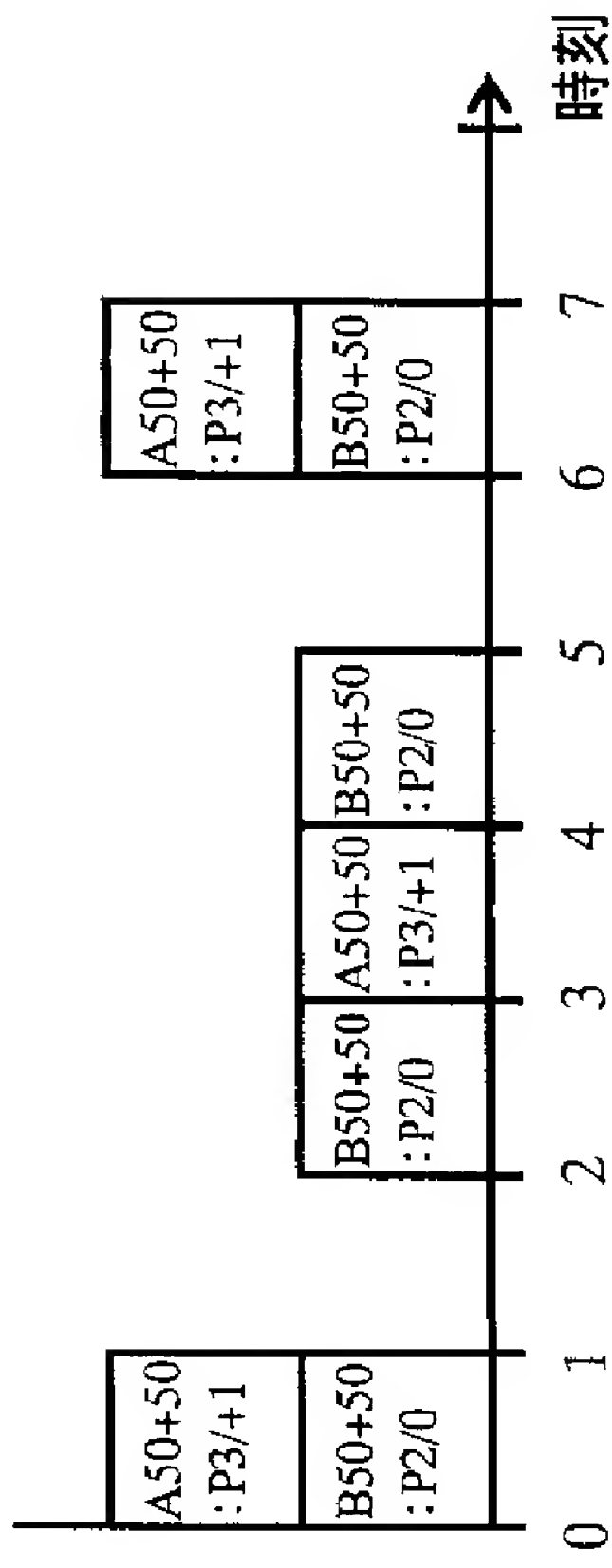
(b2) [CPU動作周波数(実行時間決定後)]



(a1)

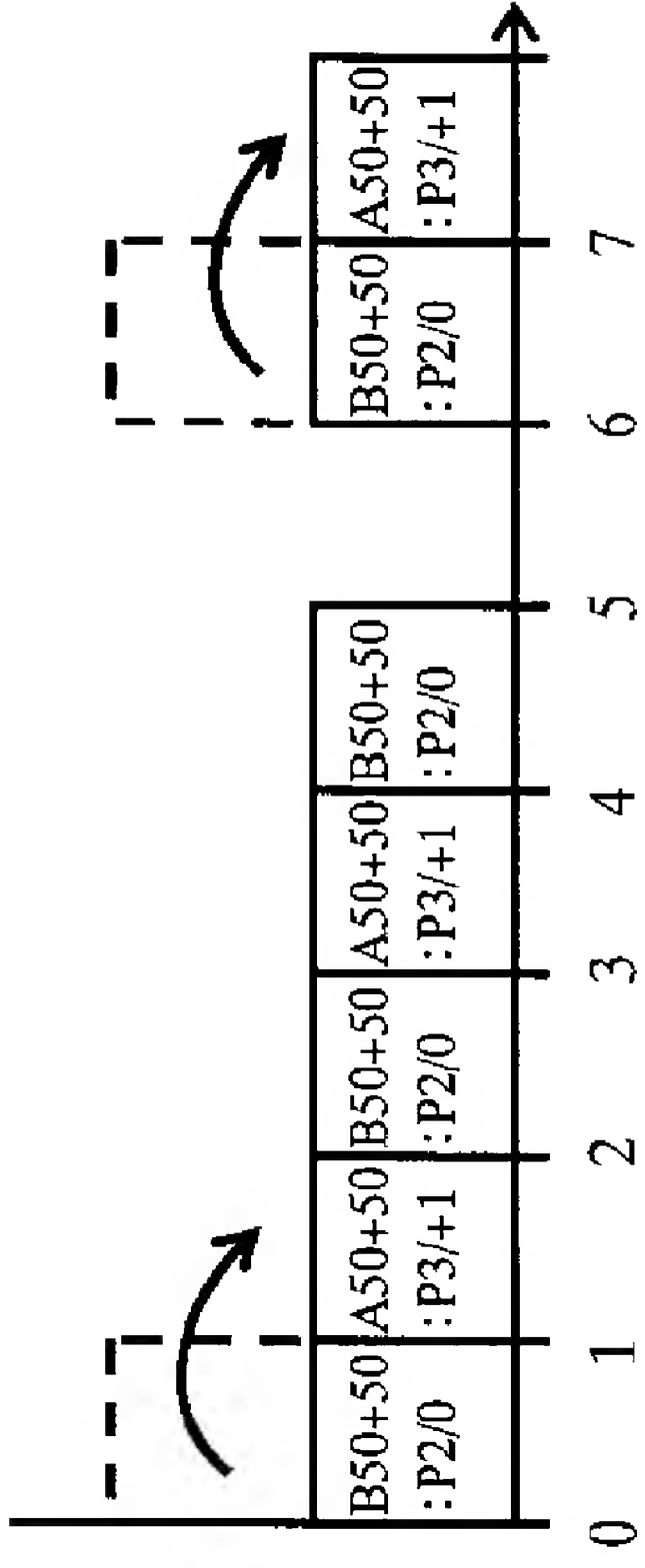
[タイマイベント登録時の実行時間要求]

必要処理量

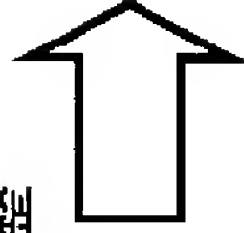


(a2)

[決定されたタイマイベント実行時間]

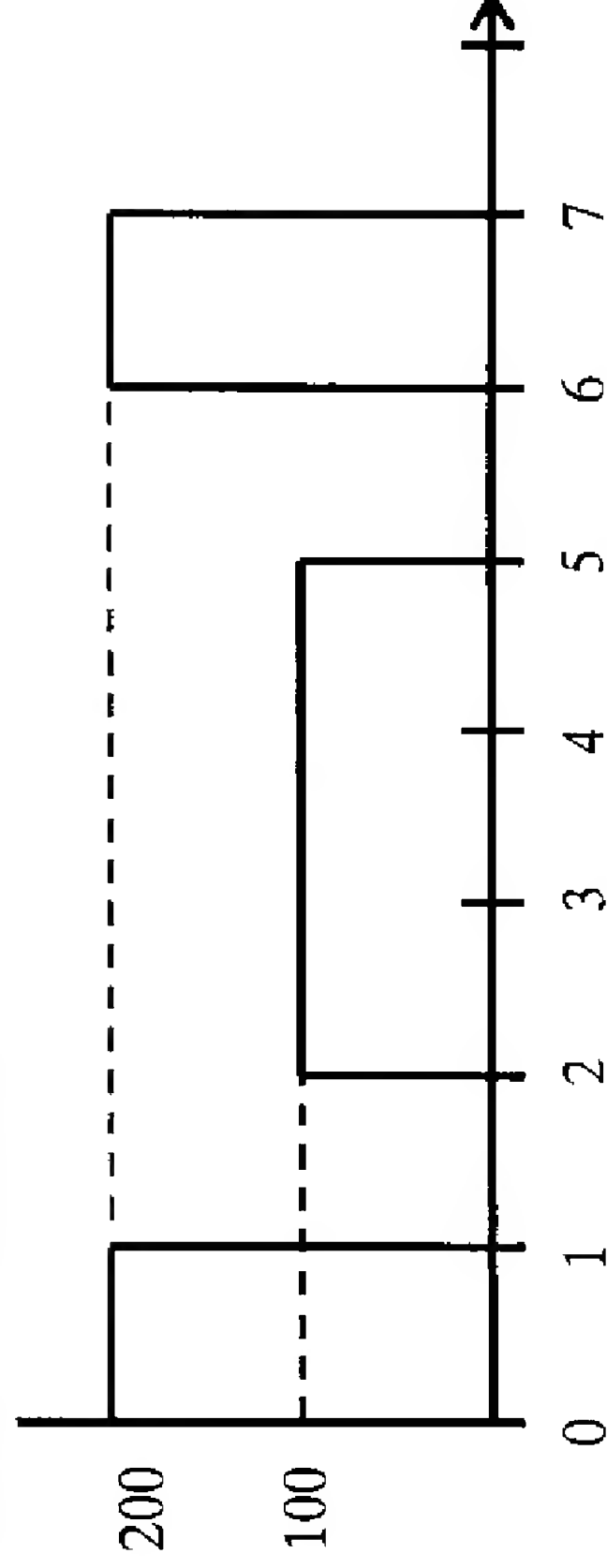


プログラム実行時間
の調整

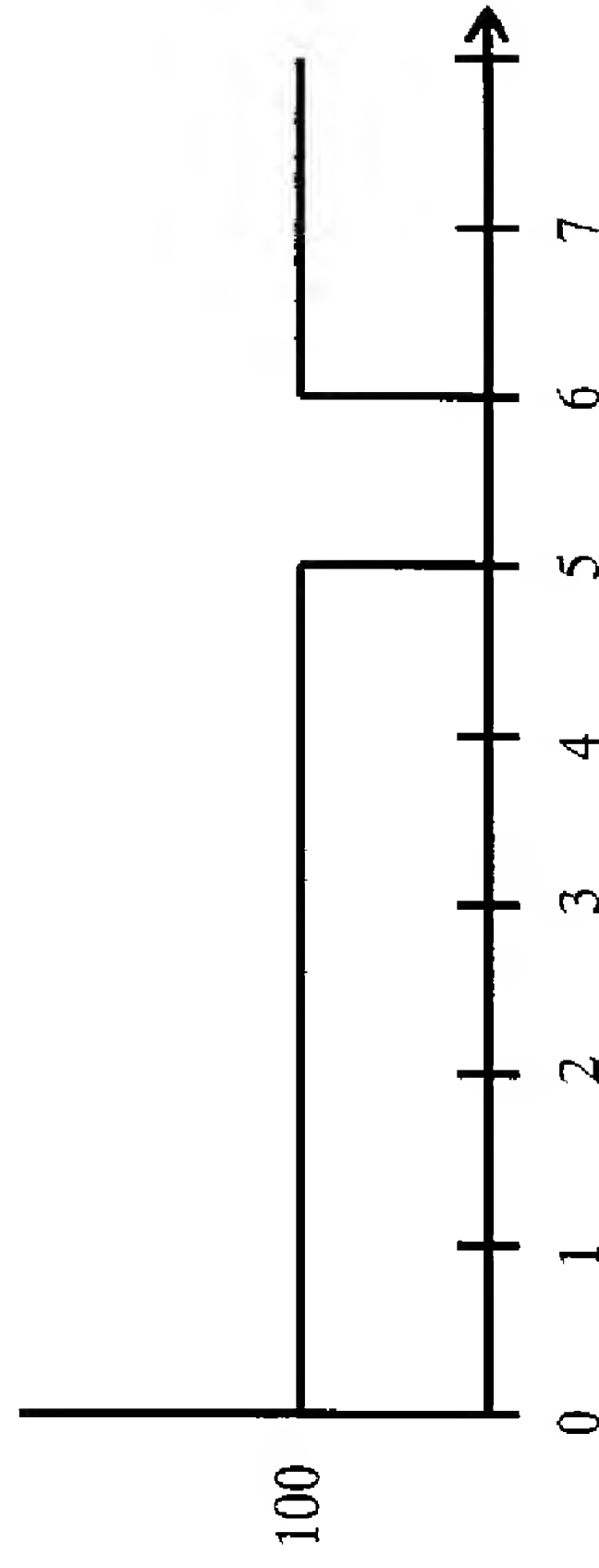


(b1) [CPU動作周波数(要求のまま実行した場合)]

動作周波数
(MHz/単位時間)



(b2) [CPU動作周波数(実行時間決定後)]



【書類名】 要約書

【要約】

【課題】 要求される処理の実行時間に関する要求を満たしながら、一層のCPU消費電力の削減と電力の有効な活用を行なうこと。

【解決手段】 登録された実行時間に関する要求と許容範囲を満たす範囲で、プログラムの実行に必要な処理量の単位時間毎の合計が均等化するように、プログラムの実行時間を決定するプログラム実行時間決定部を備える。これにより、プログラムの実行時間に関する要求の許容範囲内で必要とする処理量を極力均等化するプログラムの実行時間を決定し、CPUの動作周波数の変動を抑えるクロック制御を行う。

【選択図】 図 1

出願人履歴

0 0 0 0 0 5 8 2 1

19900828

新規登録

大阪府門真市大字門真 1 0 0 6 番地

松下電器産業株式会社